Interworking Public and Private ATM Networks

by

C. Brit Gould

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Engineering and Computer Science and

Master of Engineering in Electrical Engineering and Computer Science at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1996

Author _____
Department of Electrical Engineering and Computer Science
May 28, 1996

Certified by _____
Dr. Steven Finn
Principal Research Scientist, LIDS
Thesis Supervisor

Accepted by _____
Prof. F. R. Morgenthaler
Chairman, Department Committee on Graduate Theses

Acknowledgments

First off, I would like to thank Susan Aspie and Richard Baughman of Bellcore for helping me find a meaningful topic for my thesis and for making a lot of the work that went into this thesis possible. Also, thanks to Surinder Jain and rest of the people at Bellcore who provided useful comments and feedback, and shared knowledge along the way.

Many thanks go to Dr. Steven Finn, who supervised this thesis, helped me bring all the sections together for sensible organization, and read it several times cover to cover to offer valuable feedback. Thanks also to all the faculty and staff of M.I.T. who have taught me much of what I have learned in the past few years.

Warm thanks to my friends, who were always there throughout my years at M.I.T. offering support, friendship, good advice, mutual learning, and good times, that helped me enjoy my time at M.I.T. so much, and helped me get a great deal out of the five years I spent here. Thanks also to NBC for broadcasting *Friends* on Thursdays, which we watched almost religiously.

Thanks to my uncle Edward Mackauf for suggesting I look into ATM in the first place, and to my grandmother Ethel Blum-Dublin who always tells me I can do anything I want to do.

Finally, a very special thank you to Carol Blum-Mackauf, Stephen Mackauf, Brad Gould, and all the rest of my family that means very much to me and is so important. They have made so much of this possible and have brought me so far with their support, encouragement, and so many other great things they have done for me.

Interworking Public and Private ATM Networks

by

C. Brit Gould

Submitted to the

Department of Electrical Engineering and Computer Science

May 28, 1996

In Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Electrical Engineering and Computer Science and

Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Public ATM networks are starting to be deployed by local and long distance telecom companies. Private networks are also beginning to spring up quickly. The addressing used in public and private networks, however, is not compatible. For public and private networks to interoperate a translation is needed somewhere in the network. This thesis looks at options for where in the network this translation may be done and finds the public network to be an appropriate place. It then presents an Intelligent Network based method for ATM address translation to be done in the public network. Changes in narrowband Intelligent Networks necessary to support this broadband service are discussed. A "quick-fix" for using existing narrowband Intelligent Networks for the translations is described.

Thesis Supervisor: Steven Finn

Title: Principal Research Scientist,

MIT Laboratory for Information and Decision Systems

Contents

List of Figures

1. Introduction

Asynchronous Transfer Mode (ATM) is an important part of today's emerging field of high speed networking. ATM networks are designed to carry a complete range of user traffic including voice, data, and video. ATM is the first standards based technology which has been designed from the beginning to accommodate the simultaneous transmission of data, voice, and video. ATM is very well suited for high speed networking in the 1990s and will very likely facilitate the introduction of multimedia services that have been getting so much attention 0in the past few years.

Public ATM networks, operated by Local Exchange Carriers (LECs) and Inter Exchange Carriers (IECs), are starting to be deployed. Private ATM networks are springing up even more quickly than public networks. There is a need for both public and private networks to interoperate. Public and private ATM addresses, however, are different and not compatible with each other. Public ATM addresses are ITU-T E.164 addresses while private addresses are ATM End System Addresses (AESAs) defined by the ATM Forum.

ATM Forum specifications have been defined to provide only limited interoperability between public and private ATM networks. In order for public and private ATM networks to interoperate efficiently, a translation is necessary so that calls to and from private networks may traverse the public network and vice versa.

1.1 Research

This thesis first collects knowledge about the "state of the art" in network addressing and addressing scheme interworking. Then, analyzing how ATM calls may be set up across networks, translation in the public network is found to be appropriate. Two feasible methods for translating between private and public ATM addresses are found to be an

Intelligent Network based model, and the other modeled after the Internet Domain Name System.

This thesis studies how Intelligent Networks can be used to offer an ATM address translation service to users of an ATM internetwork of public and private networks, so as to facilitate interoperability between public and private ATM networks.

The address translation service presented in this thesis is based on existing narrowband Intelligent Networks. Changes that must be made for these Intelligent Networks to support broadband are discussed. Both long term and short term solutions are discussed. A short term solution is one where minimal changes are made to existing narrowband Intelligent Networks in order to deploy the service quickly, before broadband Intelligent Networks are standardized.

1.2 Organization

The basics of ATM are introduced section 2 and Intelligent Networks are described in section 3 as background information for the thesis. They are intended to familiarize readers with ATM and Intelligent Networks. Section 4 describes ATM addressing as well as other common addressing schemes used in today's common networks. Section 5 then discusses how these addresses are translated in order to understand the "state of the art" in the field of network address translations.

Section 6 addresses the issue of interworking public and private ATM networks. Four common topologies for interconnecting public and private networks are studied to understand the signalling necessary to set up a call across networks in several configurations and to see where ATM address translations are needed. This discussion is followed by an analysis of where the translation should be done in the internetwork.

Having found translation in the public network to be most appropriate, section 7 discusses options for carrying out ATM address translation in the public network. Then section 8 explores how an ATM address translation service can be done with Intelligent Networks. Changes that would have to be made to the existing narrowband Intelligent Network to support broadband are discussed in this section.

Finally, conclusions are presented in section 9.

1.3 A Note About References

A significant amount of material referenced in this thesis comes from standards bodies and a great deal of this material can be found on-line through the World Wide Web.

The ITU is the International Telecommunication Union. The ITU is an international organization within which governments and the private sector coordinate global telecom networks and services. ITU activities include the coordination, development, regulation and standardization of telecommunications. References in this thesis come from the ITU-T which is the Telecommunication Standardization Sector of the ITU. The ITU Web site can be found at "`http://www.itu.ch/`".

The ISO is the International Organization for Standardization. The ISO is a worldwide federation of national standards bodies from about 100 countries, one from each country. The ISO Web site can be found at "`http://www.iso.ch/`".

ANSI is the American National Standards Institute. ANSI is the U.S. representative to the ISO. Its Web site can be found at "`http://www.ansi.org`".

The ATM Forum is a worldwide organization, aimed at promoting ATM within the industry and the end user community. The ATM Forum Technical Committee works with other standards bodies and with its over 700 member companies in order to promote a single set of specifications for ATM products, thus ensuring interoperability between vendors and products. Their Web site can be found at "`http://www.amtforum.com/`".

RFCs referenced in this thesis are Internet Requests for Comments. RFCs describe the Internet suite of protocols and related experiments. Not all RFCs describe Internet standards, but all Internet standards are written up as RFCs. RFCs for standards normally originate as proposed standards or proposed protocols for the Internet. RFCs can be upgraded from proposals to standards. Their name, however, remains RFC which can be deceiving. RFCs can be found on-line through the Internet Engineering Task Force (IETF) at "`http://www.ietf.cnri.reston.va.us/home.html`" and can also be found directly at "`http://www.cis.ohio-state.edu:80/hypertext/information/rfc.html`".

Finally, Bellcore documents referenced are standards for telephony in the U.S. Telecom standards in the U.S. often either come out of Bellcore and are then adopted by other standards bodies such as the ITU-T, or they come out of standards bodies and are then adopted by Bellcore. Bellcore's Web site can be found at "`http://www.bellcore.com/`".

2. Asynchronous Transfer Mode

Asynchronous Transfer Mode (ATM) is a key component in the emerging field of high speed networking. ATM is a telecommunications concept defined by ANSI and ITU-T standards for carrying a complete range of user traffic including voice, data, and video, over a standard User-to-Network Interface (UNI). ATM is very well suited for high speed networking in the 1990s. ATM technology can be used to combine user traffic from existing applications onto a single UNI, and to facilitate multimedia networking between high speed devices at multi-megabit speeds [6]. ATM was formally adopted in the late 1980s by the standards committees as the underlying transport technology for Broadband Integrated Services Digital Networks (B-ISDNs).

This section provides an overview of ATM intended specifically to familiarize readers with ATM technology.

2.1 The B-ISDN Protocol Reference Model

The basic model of ATM networks is represented by the B-ISDN protocol reference model defined by the ITU [6,13]. This reference model is divided into multiple planes, as shown in Figure 2-1 on the following page.

The User plane (U-plane) provides for the transfer of user application information. It contains the Physical layer, ATM layer, and multiple ATM Adaptation Layers (AALs) required for different service users (e.g. CBR service, VBR service). The AALs insulates the upper layers of the user's application protocols from the details of the ATM transport mechanisms. AAL supports segmentation and re-assembly of higher layer (large) packets, possible error checking and multiplexing.

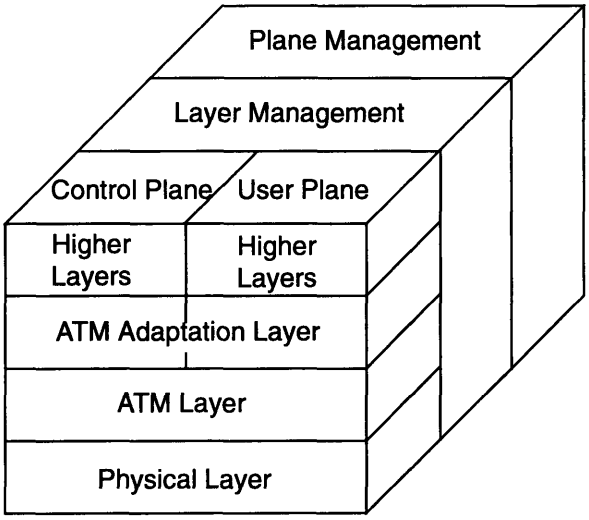


Figure 2-1. B-ISDN Protocol Reference Model

The Control plane (C-plane) protocols deal with call establishment and release and other connection control functions necessary for providing switched services. The Control plane shares the Physical and ATM layers with the User plane. The AAL used by the control plane is SAAL (Signalling AAL).

The Management Plane (M-plane) provides management functions and the capability to exchange information between the User and Control planes. The Management plane is divided into two sections: Layer Management and Plane Management. Layer Management performs layer-specific management functions while Plane Management performs management and coordination functions for the complete system.

2.2 Statistical Multiplexing

To describe statistical multiplexing, a brief introduction to Synchronous Transfer Mode (STM) may be helpful. STM is used in more conventional networks to transfer packets, usually over long distances. A connection between two endpoints is set up before packets are exchanged and torn down when they are done. The network allocates and reserves a certain bandwidth for the entire duration of the call, whether or not data is being transferred. The total bandwidth of an STM link (such as a T1 or T3 link) is divided into units of transmission: time-slots. On a given STM link, a connection between two endpoints is assigned a fixed time-slot. Data from that connection and only from that connection is always carried in the assigned time-slot. If a time-slot is reserved but not used at any given time it is simply wasted bandwidth. The total number of connections active on any given link can be at most equal to the number of time-slots.

Statistical multiplexing, used by ATM, attempts to solve the problem of unused time-slots. Several connections may be "statistically" multiplexed on the same link. For example, if there are a large number of bursty connections, they may all be assigned to the same link with the hope that statistically they will not burst at the same time. By using statistical multiplexing, the sum of the peak bandwidth requirement of connections on a link can exceed the aggregate bandwidth of the link.

While it is possible for multiple connections to use any available bandwidth, it is also possible in ATM to reserve a certain amount of bandwidth for a connection. The difference between reserved bandwidth in an STM network and an ATM network is that in ATM while the reserved bandwidth is guaranteed (as it is in STM), the unused bandwidth can be used by other connections.

2.3 Virtual Circuits

Most links between peers can be described by one of three models: *circuits*, *datagrams*, and *virtual circuits*. Telephone networks (POTS, or plain old telephony service) use circuits, connectionless networks such as IP networks use datagrams, and ATM networks use virtual circuits. Virtual circuits have the benefits of statistical sharing of bandwidth (like datagram networks), and have the benefit of reserved resources to guarantee quality of service (like physical circuits). To establish a virtual connection link, the destination address need be specified only once, during circuit establishment. A route is then set up through the network, and resources are allocated to carry messages between the specified endpoints. The endpoints and midpoints along the path identify this route with a *virtual channel identifier*. This virtual channel identifier is attached to each message sent along the previously established route to the call's destination.

The terms *connection oriented* and *circuits* are often used together. This is because with virtual circuits a connection must first be set up before data can be transferred. This may be easiest to understand when contrasted with *datagrams*, which are normally *connectionless*. Datagrams carry the destination address so that each datagram can be routed individually — no previous connection establishment is necessary or used at all.

In fact, a datagram network layer may be implemented over a connection oriented network layer. An example of this are implementations of IP over ATM, such as those described in section 5.

Some advantages of virtual circuits are that once a connection is set up, and endpoint only needs to specify the virtual channel identifier and the data is sent in-order to the destination. The overhead in the packet is therefore reduced and in-order delivery is guaranteed. On the other hand, datagram service is generally easier to provide. It requires less resources to be

dedicated to the *state* of a circuit or route through nodes connecting endpoints. This is achieved by pushing responsibility for sequencing farther up the protocol stack of the endpoints. [42]

Each ATM cell (or packet of data) carries a virtual connection identifier. This identifier is subdivided into two fields: the Virtual Path Identifier (VPI) and the Virtual Channel Identifier (VCI). The ATM network uses the information in the VPI/VCI pair to forward each cell to the proper destination. This VPI/VCI pair identifies a Virtual Channel Connection (VCC). Each VPI is conceptually a bundle of virtual channels. The Virtual Path concept is illustrated in Figure 2-2 below.

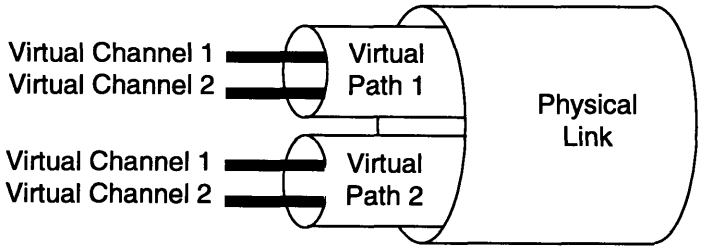


Figure 2-2. Virtual Paths and Virtual Channels

2.4 Cells

The ATM unit of transmission is the *cell.* ATM cells have a fixed length of 53 bytes. Five of these bytes are the cell header and the other 48 bytes are the cell payload. Figure 2-3 shows the ATM cell format. In general, the network carries the payload from end to end examining only the header. There are some services however, that examine the payload as well. This is done, for example, for early packet discard where if a cell is dropped all other cells are dropped until the beginning of the next packet.

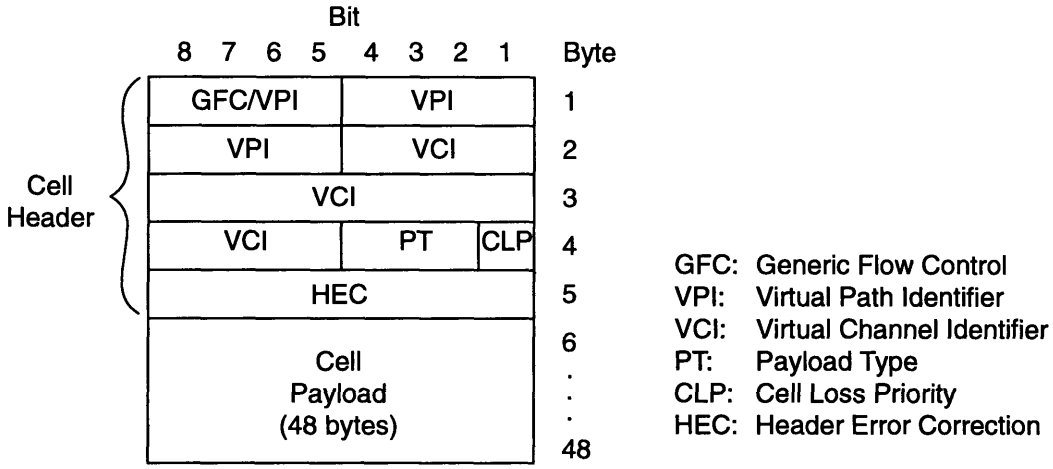


Figure 2-3. ATM Cell Format

The VPI has two defined sizes. At the Network Node Interface (NNI) it is 12 bits, and at the User Network Interface (UNI) it is 8 bits wide [14]. The VCI is 16 bits wide, allowing each virtual path to carry 65535 virtual channels.

The Generic Flow Control (GFC) field is intended to provide local functions at the customer site. Proposed functions are flow control and medium access contention resolution in environments where switch ports are shared by multiple ATM interfaces [42]. The value of the GFC is not carried end-to-end and is overwritten by ATM switches in the network.[1]

The Payload Type Indicator carries a variety of information between ATM layers and their user entities. It is used to distinguish between cells carrying user data and network

1. Note how the 4 bits that correspond to the GFC at the User-Network Interface are part of the VPI at the Network Node Interface (see Figure 2-3).

management information. It can also be used to carry network congestion and resource management information.

The Cell Loss Priority (CLP) can be set by the cell's originator or any switching node along the path of the virtual connection. Cells with CLP set to 1 will be discarded first if there is congestion in the network.

The Header Error Check (HEC) is an 8 bit CRC over the entire header, not the payload [6]. The HEC can provide single-bit error correction and multi-bit error detection.

2.5 Switching

Since ATM networks are switched networks, the most important entities in the network are the switching nodes. They enable all virtual connections. Switching nodes (or switches) accept ATM cells from one virtual link and forward them on another virtual link to get the cells to their destination. This is done based on the VPI/VCI in the cell's header and the physical link it arrives on.

The switch maintains a table with mappings of VPI/VCIs. When a cell arrives, the switch looks at the VPI/VCI in the header and looks it up in its internal table. It then changes the VPI/VCI as appropriate and forwards the cell on the appropriate physical link.

ATM networks may handle virtual paths and virtual channels in two different ways. For the top four channels in Figure 2-4, the network is providing virtual channel service. The network examines both the VPI and the VCI of each cell to determine how to forward the cell towards its destination. The values of both the VPI and VCI change as the cell traverses the network. [43]

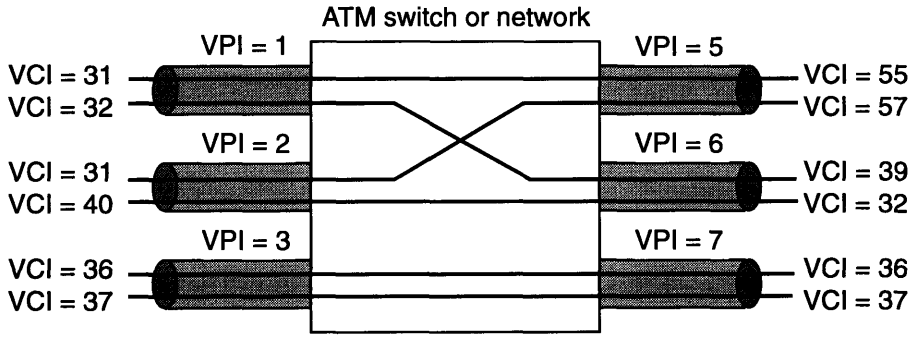


Figure 2-4. Virtual Paths and Virtual Channels in the Network

For the bottom two channels in Figure 2-4, the network is providing virtual path service. In this case, the ATM network makes the cell forwarding decision based only on the value of the VPI. The VCI does not change as the cell traverses the network.

The virtual path concept is useful, for example, in wide area networking to create a Virtual Private Network (VPN). A virtual path can be bought (or leased) from a wide area carrier for communications between two locations. Virtual channels can then be set up and torn down without having to coordinate with the carrier [43].

2.6 Quality of Service

ATM connections have an associated group of parameters that define the Quality of Service (QoS) expected. QoS involves characteristics such as peak and average cell rates, burstiness, cell loss probabilities, end-to-end delay, and cell delay variance. [42]

The relative importance of these characteristics depends on the type of traffic on a particular connection. For example, packet services will likely be more concerned with peak and average bit-rates, while a constant bit rate (CBR) video connection will likely require known bounds on overall link delay and cell delay variance.

2.7 ATM Adaptation Layer (AAL)

The ATM Adaptation Layer (AAL) is essential to services offered by ATM networks. The AAL exists at the endpoints of virtual connections, where higher layer peers wish to establish communication, and normally resides in terminal equipment. The AAL converts arbitrarily formatted information supplied by the user into an ATM cell stream, and vice versa [47]. The primary sources for information on AAL services are ITU-T recommendations I.362 [15] and I.363 [16].

Rather than developing separate AALs for each type of telecommunication service offered, several common factors between services have been drawn together to form a small set of AAL protocols. Several broad classes of AAL service are identified by the ITU-T. These cover possible combinations that may need to be supported. Criteria for the division in classes are *timing relation*, *bit rate*, and *connection mode*. For example, video services, which require the image sequence and timing at the destination to be the same as at the source, require *timing relation*, while packet data services do not. *Bit rate* may be constant or variable. The *connection mode* may be connection-oriented or connectionless.

AAL type 1 provides circuit transport to support synchronous (e.g. 64KBit/s) and asynchronous (e.g. 1.5, 2 MBit/s) circuits. It can be used for constant bit rate services such as interactive and distributed video services, voice, and high quality audio transport.

AAL type 2 has not yet been defined, but it is envisaged to support variable rate services.

AAL type 3/4 comes from a combination of AAL3 AAL4. AAL type 3 was designed for connection-orientated data, while AAL4 was designed for connectionless-orientated data. Realizing that there were almost no functional differences, they were merged to form AAL 3/4.

AAL type 5 is designed for the same class of service as AAL 3/4, but contains less overhead. It shifts some of the responsibility, such as multiplexing, up to a higher layer. AAL5 seems to be winning over AAL3/4 as the preferred AAL for connection oriented and connectionless services.

2.8 Signalling

ATM is connection oriented. As such, users must be able to rapidly create and remove virtual connections between each other "on demand." Users do this by means of a signalling protocol. The ITU-T has developed the Q.2931 signalling protocol (previously called Q.93B), a B-ISDN variant of Q.931 for narrowband ISDN. Before Q.2931 was available only proprietary signalling systems were used, but now that Q.2931 has been completed it is achieving widespread use and acceptance in the industry.

Signalling in ATM Networks allows for setting up, maintaining and releasing ATM virtual channel connections. It also allows negotiation of the traffic characteristics of a connection. The most common type of connection used in ATM is a point-to-point connection. However, signalling functions may also support point-to-multipoint or even multipoint-to-multipoint connection calls. Signalling messages are conveyed "out of band" over dedicated signalling virtual channels.

2.8.1 User-Network Interface (UNI)

The signalling protocol of the ATM Forum's UNI is based on ITU-T's Q.2931. UNI version 3.1 is the most recent version of the specification. UNI 3.1 adds to Q.2931 in the areas of point-to-multipoint connections, additional traffic descriptors, and private network addresses. UNI 4.0, which will include support for multipoint-to-multipoint connections, is up for "straw ballot" by the ATM Forum in June, 1996. Unless specified, references to the UNI in this thesis are to UNI 3.1.

The UNI specifies procedures for dynamically establishing, maintaining, and clearing ATM connections at the User-Network Interface. The procedures are defined in terms of messages and the information elements (IEs) used to characterize the ATM connection and ensure interoperability [6]. Call establishment, call clearing, and status messages are defined. Because of the scope of this thesis we will mainly be concerned with the initial call establishment message: the SETUP message.

The SETUP message can be sent from the calling user to the network, and from the network to the called user to initiate call establishment. Messages sent across the UNI carry a variety of information elements to convey information. All messages contain a Protocol Discriminator information element (to indicate that the messages are Q.2931 user-network call control messages), a Call Reference information element (to identify the call to which the particular message applies), a Message Type information element (to indicate the type or function of the message), and a Message Length information element (identifying the length of the contents of the message).

In addition, the SETUP message contains information elements necessary for initial call setup. The information elements we will focus on are the *Called Party Number* and *Called Party Subaddress*. The *Called Party Number* identifies the called party of the call. It

supports both public and private ATM addresses (these addresses are described in section 4. The *Called Party Subaddress* identifies the subaddress of the called party of a call. In UNI 3.1 it is used only to convey a private ATM address across a public network which supports only public ATM addresses.

The *Calling Party Number* and *Calling Party Subaddress* are also defined to convey information about the calling party. Their formats are the same as those of the called party information elements.

2.8.2 Network-Network Interface (NNI)

Network-network Interfaces, also known as Network Node Interfaces (NNIs) are used between switches, or nodes, in a network. Private ATM networks plan on using the Private Network-Network Interface (P-NNI) [8], while public ATM networks will use B-ICI [7], which is based on ITU-T B-ISUP (Broadband ISDN User Part) [23,24,25,26]. B-ICI version 2 is the most recent version.

Since this section is intended as an overview of material necessary for understanding later sections of this thesis, only B-ICI (the public NNI) is discussed here.

The counterpart to the UNI SETUP message in B-ICI is the Initial Address Message (IAM). The IAM is sent from one switch to the next in the path of a call to initiate call setup, much like the SETUP message is sent from the user to the network or from the network to the user across the UNI. The IAM is the first message used in "forward address signalling" in B-ICI.

The IAM contains all the information required to route the call to the destination. The IAM contains the *Called Party Number*, and in some cases the *AESA for Called Party* parameters, as well as traffic, quality of service, and other parameters.

The *AESA for Called Party* and *AESA for Calling Party* parameters were recently defined by the ITU-T in draft Recommendation Q.2726.1 [22]. These parameters are used to carry private ATM addresses received in SETUP *Called Party Number* and *Calling Party Number* information elements transparently across a public network.

It should be noted that although the structure of the *AESA for Called Party* and *AESA for Calling Party* parameters would allow them to carry any of the three types of private ATM addresses (AESAs), the draft Recommendation specifies that they will only be used for AESAs in E.164 format. Currently, some members of the ITU want to limit the use of these parameters to E.164 AESAs, as in the current state of the draft, while others would rather open it up to allow carrying ICD and DCC AESAs through the public network as well as E.164 AESAs. The direction of this issue in the near future is not clear at this time. However, the draft Recommendation is now frozen, and will likely be approved in 1996.

3. Intelligent Networks

The Intelligent Network (IN) is a service-independent telecommunications network architecture. It provides capabilities for the rapid creation of customizable telecommunications services by network and service providers and their customers. IN provides a framework to introduce, control, and manage services efficiently, economically, and quickly. The primary objective of IN is to reduce the time required to define, develop, and deploy new telecommunications services.

In this section some of the basics of Intelligent Networks are described, so as to familiarize readers with Intelligent Networks. The concepts of Intelligent Networks presented here are needed to better understand later sections in this thesis.

The Intelligent Network separates call control from connection control. The connection of calls is handled by switches in the network, while services are handled by Service Control Points (SCPs) in the network. SCPs are described below.

Since the Intelligent Network was originally designed for telephony (narrowband) networks, the next few subsections discuss INs used today in telephony. Section 3.6 then gives an overview of how IN can apply to broadband.

3.1 The World Before IN

Around 1965, stored-program control telephone switches were introduced. Call processing, billing, and service logic functions were all carried out by software in the switch. This allowed for new originating services (services at the originating end of a call) such as three-way calling and speed dialing, and new terminating services (at the terminating end of a call) such as call waiting and call forwarding.

In 1976, Common Channel Signalling (CCS) was introduced for trunk signalling — signalling between switches in the network. With CCS, all network signalling is sent *out-of-band*, and only the voice portion of the call travels through the trunk lines. User to network signalling may still be *in-band* (e.g. dial tone, touch tones, etc.). Figure 3-1 shows a telephone network with its Common Channel Signalling network. The STPs (Signalling Transfer Points) handle "switching" for the signalling network.

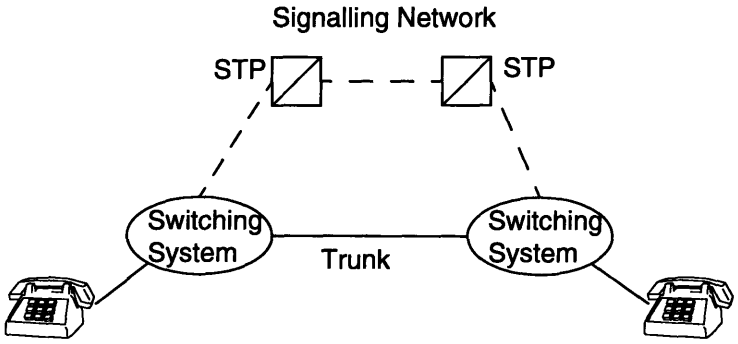


Figure 3-1. Common Channel Signalling

With these networks in place, Intelligent Networks came along.

3.2 IN Architecture

Figure 3-2 shows the elements of an Intelligent Network.

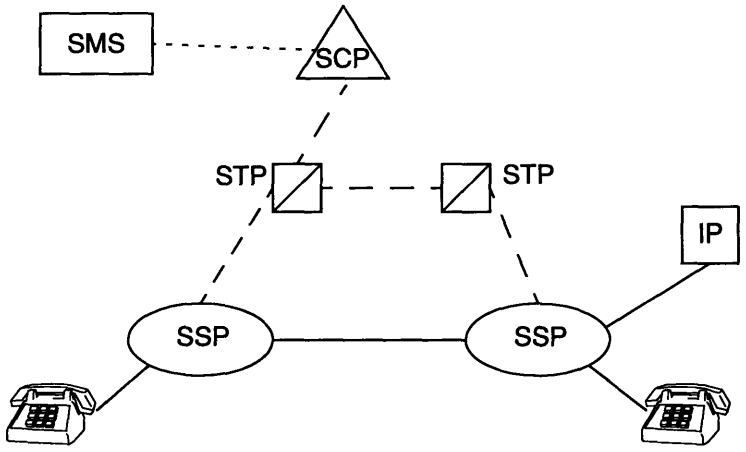


Figure 3-2. An Intelligent Network

Service Switching Points (SSPs) are switches in the Intelligent Network. These switches perform basic, generic call processing. They follow the IN call model discussed in section 3.3. At certain points in call processing the SSP may hit *triggers*, indicating that additional service processing is required. At these points the SSP normally queries the Service Control Point (SCP) to determine how to continue call processing. The SCP then replies, instructing the switch on how to continue processing the call with the service requested. SSP capabilities and triggers are service independent.

Service Control Points (SCPs) contain service independent capabilities used to support multiple services. SCPs contain service logic that is run to offer services to customers. SCPs often contain databases needed for the services they offer. Some SCP services, such as 800 number service and calling card services, require large amounts of data. This data is normally stored on disks on the SCP.

Signalling Transfer Points (STPs) are part of the CCS network. The CCS network used today is known as SS7 (Signalling System 7), named after the signalling protocol currently used [1]. STPs switch SS7 messages between SSPs and SCPs.

Service Management Systems (SMSs) update the SCP with new data and service logic and collect statistics from the SCPs. The SMSs also enable customers to control their own service parameters.

All signalling travels over the SS7 network, shown with dashed links in Figure 3-2, while voice connections are established between SSPs as shown by the solid links in the figure. Note that SCPs sit on the SS7 network and are therefore can only be accessed by switches, not directly by customers.

Intelligent Peripherals (IPs) interact with users of certain services. IPs are often used for speech recognition, speech synthesis, digitized speech recording and playback, among other services.

3.3 The IN Call Model

IN calls follow a specific call model. The IN call model consists of two half call models: the originating and a terminating half call models. Every switch along the route maintains an originating and terminating call portion.

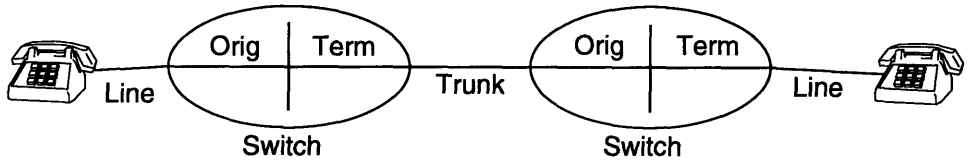


Figure 3-3. IN Basic Call Model

Both the originating and terminating call models define a set of call processing events that can be trapped and used to invoke IN services. The basic call model consists of:

- Points in Call (PICs), which represent normal call processing states in the switch.
- Transitions between PICs.
- Detection Points (DPs), where triggers can be set to detect events in the call. At these points the switch can send an SCP notification of a triggered event and the IN service logic can assume control.

Rather than discussing the entire originating and terminating call models here, we will focus on the first few steps of the originating call model in order to serve as an example of how the call model works and to provide the necessary background for later sections of this thesis. The call model described is that of Bellcore's Advanced Intelligent Network (AIN).[1] The first 5 Points in Call (PICs) of the originating call model are shown in Figure 3-4. PICs in the figure above are shown as rectangles (O_NULL through SELECT_ROUTE). Detection points are shown as squares (Origination Attempt, Info Collected, etc.).

1. The Advanced Intelligent Network (AIN) is the specific version of IN used by telephone networks in the U.S. [3,4]

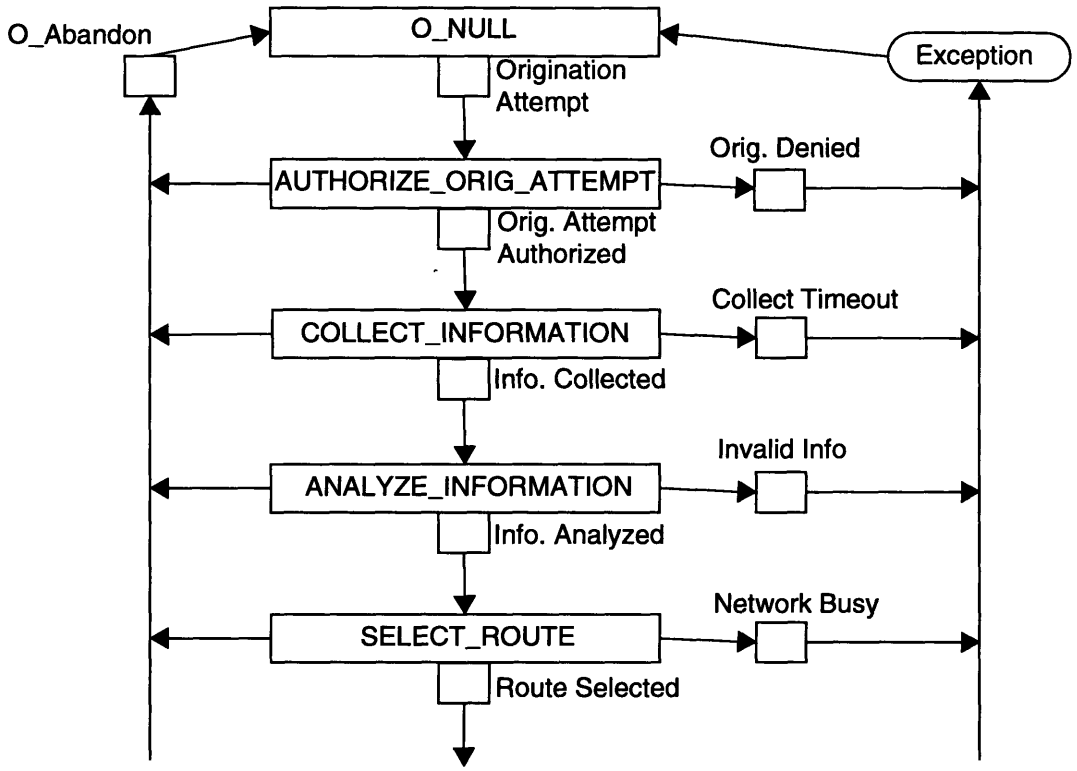


Figure 3-4. IN Originating Call Model (first 5 PICs)

The originating portion of the call model starts at the O_NULL PIC. When a user goes "off hook" the call proceeds into the AUTHORIZE_ORIG_ATTEMPT PIC. Here authorization is checked, to see that the user is allowed to make calls.

If the user is authorized to continue, a dial tone is given and call processing continues in the COLLECT_INFORMATION PIC. The user dials digits and the switch collects them. The *Info Collected* trigger can be set in this PIC for the switch to query an SCP after the dialed number has been collected. If the trigger is hit, the switch halts call processing, sends a message to the SCP and waits for a response.

If no triggers are hit, call processing continues in the ANALYZE_INFORMATION PIC. In this PIC the switch analyzes the collected digits. The *Info Analyzed* trigger can be set in this PIC to detect a "specific digit string" in the collected digits. This can be used, for example, to detect if the user dialed an 800 number.

Next is the SELECT_ROUTE PIC. In this PIC the switch selects the route to take based on the called party number. The called party number may be either the number dialed by the user, or a number sent back in a response from the SCP. The rest of the call model deals with getting the call to its destination after the route is chosen.

3.4 AIN TCAP Messages

IN messages referred to in this thesis are those of AIN specifically.

When an *Info Collected* trigger is hit during call processing, the switch sends an *Info_Collected* TCAP message to the SCP. TCAP is the Transaction Capabilities Application Part of SS7. It is the protocol used between switches and SCPs in Intelligent Networks. The *Info_Collected* message contains parameters such as the *CollectedAddressInfo* and *CollectedDigits* parameters. An appropriate response to an *Info_Collected* message is an *Analyze_Route* TCAP message sent from the SCP to the switch.

The *Analyze_Route* message instructs the switch to route the call to the party identified in its *CalledPartyID* parameter.

When an *Info Analyzed* trigger is hit during call processing, the switch sends an *Info_Analyzed* TCAP message to the SCP. The *Info_Analyzed* message contains parameters such as *CalledPartyID*, *CollectedAddressInfo*, and *CollectedDigits*. An

appropriate response to an *Info_Analyzed* message is also an *Analyze_Route* TCAP message sent from the SCP to the switch.

In the remainder of this subsection several parameters worth mentioning are described briefly. *AINDigits* is a format used by several parameters that include digits as part of their contents. This format allows a parameter to carry digits and an indication of the numbering plan among other information. Each octet of digit information is shared by two digits.

The *CalledPartyID* and *ColletedAddressInfo* parameters have *AINDigits* format and are limited to carrying up to 15 digits (8 octets-worth of digits). The *CollectedDigits* parameter also has the *AINDigits* format, but this parameter can carry up to 42 digits (21 octets of digit information).

A *GenericAddressList* is a list of up to 5 *GenericAddress* parameters. The *GenericAddress* parameter has a format similar to that of *AINDigits* and can carry up to 8 octets of digits (15 or 16 digits).

The *ExtensionParameter* exists to allow networks to define new parameters that are not defined in AIN. The definition of this parameter is flexible, so as to allow anything to be defined in the *ExtensionParameter*. Since the contents of this parameter are not standard, the contents of this parameter will most likely not be processable by SSPs unless specific arrangements are made with equipment suppliers.

3.5 Putting It All Together

IN services are constructed from IN "building blocks." The basic building blocks most often used in INs are digit collection, call routing, number translation, and playing of announcements. Services can be deployed by assigning and activating certain triggers in

switches and deploying service logic programs in SCPs. For example, for 800 number service, the switch first collects digits and hits a trigger detecting that the dialed area code is 800. A message is sent to the SCP, where a service logic program translates the number. A message is then sent back to the switch telling it where to route the call. Finally, the switch routes the call to its destination.

Triggers can be provisioned on a line-by-line basis as well as on an entire end-office basis.

3.6 Broadband IN

The Intelligent Network described so far was a narrowband (telephony) network. Over the past few years telephone networks have been upgraded to AIN Release 0.1, and in some cases AIN Release 0.2 [3,4]. Some believe that a wide deployment of ATM switches by Local Exchange Carriers (LECs) and Inter Exchange Carriers (IECs) can only be economical if broadband services are offered within the framework of IN [45]. For broadband services to be offered as IN services, there has to be a specification of an IN call model for B-ISDN, and a broadband architecture to support it.

Bellcore, who developed narrowband IN, is currently working on a broadband version of IN. Details of their broadband IN, however, are not yet public. ITU-T SG 11 (Study Group 11: Signalling and Switching) is also studying alternatives for broadband IN. However, as of yet there is no standard architecture or call model.

The broadband IN call model will differ from the narrowband IN call model. Several broadband IN call models have been proposed in the last few years [45]. The proposed call models recognize that the broadband user-network interface is different than the narrowband user-network interface and that call setup signalling is also quite different. For example in telephony (narrowband) networks, one lifts the receiver, waits for a dialtone,

and then dials a phone number. Feedback is provided to the user through in-band signalling, ringing or busy signals, that indicate to the caller the status of the call. On the other hand, in broadband networks a SETUP message is sent from the user to the network containing a wealth of information, including the called party number. Feedback is provided to the user through protocol messages (with their parameters), indicating the status of the call. This means that in broadband networks, the switch has all the information it needs to set up a call by the time the AUTHORIZE_ORIG_ATTEMPT PIC is reached. Therefore the AUTHORIZE_ORIG_ATTEMPT and COLLECT_INFORMATION PICs need not be separate PICs.

The broadband IN architecture is also likely to differ from that of narrowband IN. The SS7 network, as exists today, may not be used because it is likely to introduce an appreciable delay in communication between the switch and SCP, thus slowing down call processing. In broadband IN, the Common Channel Signalling network may be a physically separate network as it is in telephony, or it may operate over a designated virtual channel of the ATM network.

These issues are currently under study by various organizations. Although the broadband IN call model, architecture, and certain messages will differ from those of narrowband IN, the parts of broadband IN that are relevant to this thesis are very similar to their narrowband IN counterparts. To the extent that this thesis uses the concepts of IN, it will assume broadband IN to be much like the existing narrowband IN. Any differences, assumptions, and alternatives are explicitly stated.

4. Addressing Schemes

This section looks at ATM addressing schemes as well as several other widely used addressing schemes to see how existing networks and internetworks deal with subnetworks and address translations.

The first two addressing schemes described in this section are the public and private ATM network addressing standards. The first, described in section 4.1, is the addressing scheme used in public ATM networks (ITU-T Recommendation E.164). The second, described in section 4.2, is the addressing scheme is used in private networks. Private ATM addresses are also known as ATM End System Addresses (AESAs). There are three different types of AESAs. Although one of the AESA formats includes an E.164 number embedded inside of it, public (E.164) and private (AESA) ATM addresses are not compatible with each other.

Sections 4.3 through 4.6 describe other common addressing schemes. IP addressing, used in the Internet Protocol, is discussed in section 4.3. Medium Access Control (MAC) addresses, used at the data-link layer in many shared-medium connectionless networks such as Ethernet, are discussed in section 4.4. IP and MAC addresses can be used over ATM by using protocols to simulate existing network environments. Two protocols that allow IP and MAC over ATM are RFC 1577 and the ATM Forum's LAN Emulation protocol, discussed in section 5.

SS7 Signalling Point Code addressing is described in section 4.5. These addresses show one of many ways in which addresses can be split up to allow for domains and subdomains in networks. SS7 addressing is used internally in SS7 networks.

Finally, X.121 addresses, used in X.25 networks, are described in section 4.6. Currently, there are no plans for using X.121 addresses in ATM. However, ATM and X.25 networks may someday be interworked.

4.1 E.164: ATM Public Addresses

ITU-T Recommendation E.164 [11] defines the numbering plan for an Integrated Services Digital Network (ISDN). Public networks in North America use E.164 addresses and it is recommended that public networks around the world use E.164 addresses as well. These addresses are used for services ranging from Plain Old Telephony Service (POTS) and faxes, to Narrowband ISDN data and voice services. Since they all use the same addressing scheme, addresses assigned to ATM endpoints in public networks would be compatible with all other E.164 addresses. In fact, ITU-T draft Recommendation E.191 defines the addressing scheme for broadband networks and is based on the E.164 numbering plan. These public ATM addresses are still referred to as E.164 addresses.

E.164 addresses uniquely identify endpoints on, or interfaces to the public network. Several E.164 numbers can identify the same interface to the public network. As NANP (North American Numbering Plan) resources, they are allocated within the public network.

An E.164 address can be up to 15 decimal digits long and consists of two fields. These fields are the Country Code (CC) and the National (Significant) Number (N(S)N). See Figure 2-1.

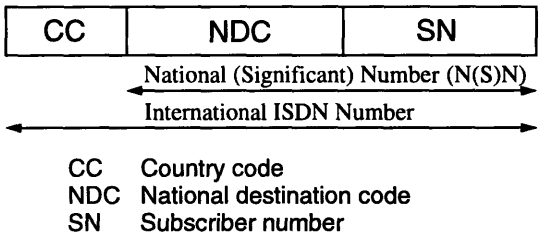


Figure 4-1. E.164 Number Structure

Country Codes (CCs) are defined in ITU Recommendation E.163 [10]. They identify countries and may vary in length from one to three decimal digits. The Country Code assigned to the U.S., for example, is 1 — the U.S. is in World numbering Zone 1.

The National (Significant) Number (N(S)N) identifies a subscriber. In identifying the destination subscriber, however, it may be necessary to identify a destination network. Therefore the N(S)N field is itself made up of two fields: a National Destination Code (NDC) followed by the Subscriber's Number (SN).

The length of the NDC field is variable depending upon the requirements of each country. In the U.S. the NDC field is 3 digits long and is known as the Numbering Plan Area (NPA), or "Area Code."

In the U.S. and in Canada (World Numbering Zone 1), for example, where the Country Code is "1" and is followed by an Area Code (NPA) the format is the familiar *1 (NPA) NXX-XXXX* used in telephony.[1] This format can also be used for ATM and in fact will be used in public ATM networks.

1. In NXX-XXXX, N is a number between 2 and 9, and X can be any number between 0 and 9.

The number assignment in World numbering Zone 1 is conducted in accordance with NANP. The NANP administrators allocate these resources (E.164 addresses) to public network operators who in turn assign them to subscribers in the public network. Some valid E.164 addresses would never be assigned by the NANP authorities due to NANP rules on assigning numbers. For example, NANP rules dictate that the first three digits (NXX) of the 7-digit subscriber number should never be the same as an Area Code (NPA) neighboring the subscriber's Area Code. Valid E.164 addresses that are not valid under NANP rules are known as "Ugly E.164" addresses. These addresses might be used by some private networks. However, interworking public networks and networks with Ugly E.164 addresses may very well pose a problem because public networks rely on NANP rules for routing (e.g., under NANP rules a number dialed that starts with a neighboring Area Code is known to be a call to that neighboring area rather than to an Ugly E.164 addressed endpoint within the local area.)

NANP originated as the addressing and numbering scheme for identifying subscribers and their location on the public switched telephone network (PSTN) in North America. There is an increasing demand for numbers with the increase in the number of home and business communication services, such as fax, computer, cellular, PCS, Internet growth, LANs, WANs, etc. NANP administrators are conservative in assigning resources, but new services and technologies are increasing the demand on these resources. In order to prevent early exhaustion of resources they are only assigned to public networks. Local Exchange Carriers (LECs) must provide nondiscriminatory access to numbers for assignment to other carriers' exchange service customers until number administration guidelines are established.

Public networks have been dealing with and routing these numbers for years. The Country Code, National Destination Code, and Subscriber Number give E.164 addresses sufficient information for global routing. These addresses are useful for organizations that wish to use

the existing largely geographically based public ISDN/telephony numbering format. These E.164 addresses would most likely correspond to public UNIs. Ugly E.164 addresses could be used by private networks as they will never be used in public networks. However, use of Ugly E.164 addresses may pose problems for public networks to which they are connected, as explained above.

In some cases the endpoint on the public network identified by an E.164 address will be a private network. This private network may use either E.164 or AESA (see section 4.2) addressing. Because public networks are likely to understand only E.164 addresses, a connection from a public network to an endpoint in a private network can be requested giving an E.164 address (i.e., the address that identifies the private network connection to the public network) and a subaddress which corresponds to the endpoint in the private network. This subaddress would be carried transparently through the public network to the private network.

4.2 ATM End System Addresses: ATM Private Addresses

ATM has its own addressing structure, defined independently from other protocols. Its address space is disjoint from that of other protocols. Other protocols can operate over ATM, while ATM and these other protocols may evolve independently.

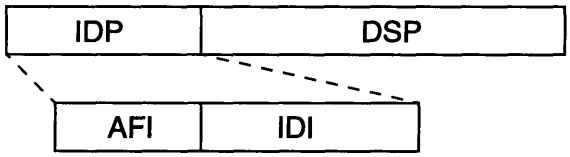
ATM private addresses, known as ATM End System Addresses (AESAs), uniquely identify ATM endpoints. They are modelled after OSI[2] NSAP (Network Service Access Point) addresses which are defined in ISO 8348 [40] and ITU-T X.213 [20] (these are the same standard, one published by ISO and the other by the ITU-T). AESAs have the same

2. OSI is the seven layer Open Systems Interconnect reference model developed by the ISO (International Standards Organization). The OSI reference model is intended to be the structure for the "ideal" network architecture.

structure, abstract semantics, abstract syntax, and preferred binary encoding as OSI NSAP addresses.

The NSAP addressing scheme follows a hierarchical structure for unique global identification of any NSAP. The structure allows allocation and assignment of NSAP addresses based on existing, well-established network numbering plans and organization-identification standards.

The NSAP address can be up to 20 octets long. It consists of two main parts: the Initial Domain Part (IDP) and the Domain Specific Part (DSP), as shown in Figure 4-2. Both of these parts may be subdivided into smaller subparts. Some commonly used lengths of IDP and DSP are explained in the next few pages.



IDP Initial Domain Part
AFI Authority and Format Identifier
IDI Initial Domain Identifier
DSP Domain Specific Part

Figure 4-2. OSI NSAP Address Structure

The Initial Domain Part (IDP) is a network addressing domain identifier. It uniquely identifies the administrative authority responsible for assigning NSAP addresses in the specified domain, i.e., the authority responsible for assigning values of the Domain Specific Part (DSP). The IDP is made up of two fields: the Authority and Format Identifier (AFI) and the Initial Domain Identifier (IDI).

The AFI specifies the authority responsible for allocating values of the IDI, and the syntax of the DSP.

The IDI specifies the addressing domain from which values of the DSP are allocated and the authority responsible for allocating these values. IDI formats for private ATM addresses are specified in ATM Forum UNI 3.1 [6].

The DSP is the corresponding address within the addressing domain specified by the IDI. The OSI NSAP specification [20][40] does not require any further substructure of the DSP. Any substructuring of the DSP is left to the organizations using the addresses and this substructure and its semantics may be defined by the authority identified by the IDP. In UNI 3.1, the DSP of AESAs is subdivided into the High-Order DSP (HO-DSP), ESI, and SEL, as shown in Figure 4-3.

Each private network may define its own HO-DSP. UNI 3.1 recommends that its definition should allow hierarchical routing and efficient use of resources, i.e. suballocation of fields within the HO-DSP should be assigned with topological significance.

The structure of the End System Identifier (ESI) and Selector (SEL), within the DSP of an AESA, are as specified in ISO 10589 [41]. The ESI is an identifier whose value must be unique within a particular value of the IDP + HO-DSP. To ensure the ability of an end system to autoconfigure its address, the ESI can be an IEEE MAC address (see section 4.4).

The SEL is not used for ATM routing but may be used by end systems, e.g., to identify an application within an end system.

The ATM Forum has proposed three AESA formats (DCC, ICD, and AESA E.164), as specified in UNI 3.1, section 5.1.3 [6]. These formats are shown in Figure 4-3 and are described in sections 4.2.1, 4.2.2, and 4.2.3 on the next few pages.

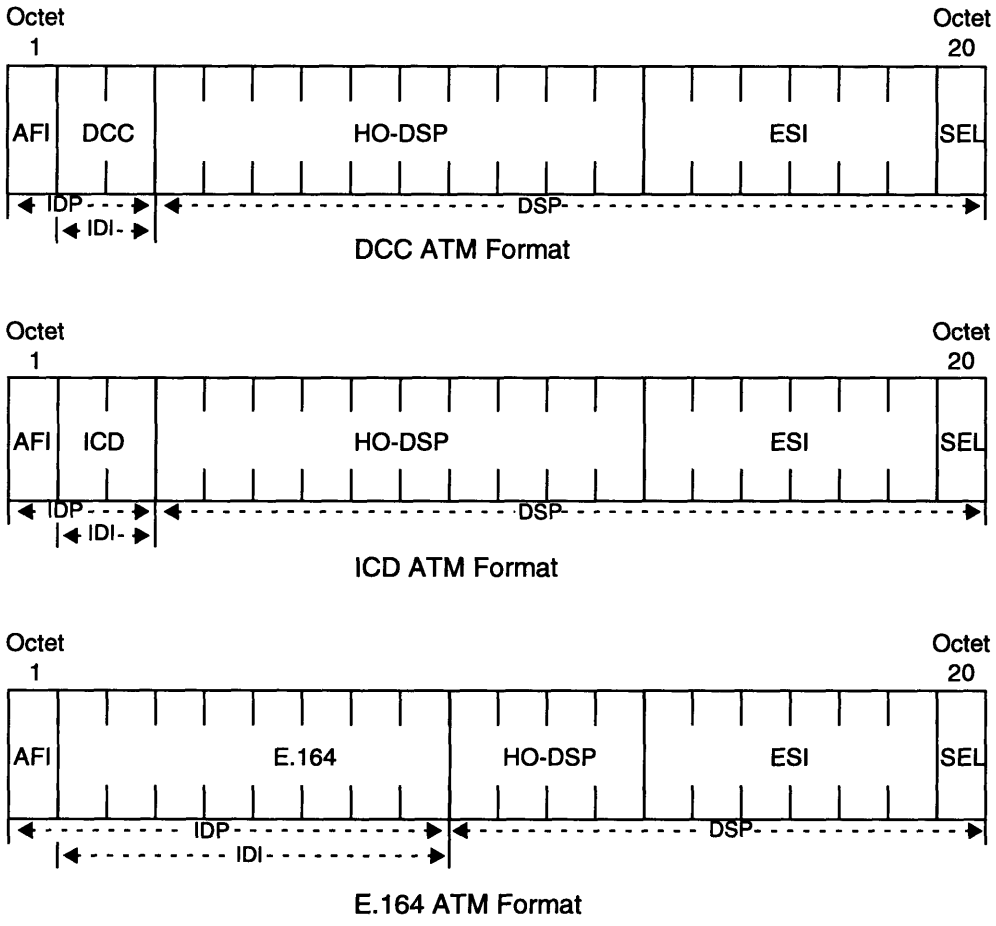


Figure 4-3. ATM End System Address (AESA) Formats

4.2.1 Data Country Code (DCC)

Data Country Codes (DCCs) identify particular countries. These codes are specified in ISO 3166 [38]. Note that these Country Codes are different from those defined in E.163 [10], which are used by E.164 [11] and telephony in general. DCCs are administered by the ISO National Member Body in each country. The Member Body in the U.S. is the American

National Standards Institute (ANSI). The length of the DCC field in AESAs is two octets (see Figure 4-3).

The DCC format is useful for organizations that wish to maintain a private numbering plan that is organizationally based on country.

4.2.2 International Code Designator (ICD)

International Code Designators (ICDs) identify particular international organizations. The ICDs are allocated by the ISO 6523 [39] registration authority, the British Standards Institute. The length of the ICD field in AESAs is the same as that of the DCC: two octets.

The ICD format is useful for organizations that wish to maintain a private numbering plan that is organizationally based.

4.2.3 AESA E.164

The AESA E.164 address (sometimes referred to as the NSAP E.164 address) is an AESA format address with an E.164 address embedded in the IDI. More information could fit in the 8 octets of IDI assigned to the E.164 number in this format than could fit in the public E.164 format which allows only up to 15 decimal digits. However, UNI 3.1 specifies that the 8 octets of IDI should be padded, leaving enough space for 15 decimal digits-worth of information to represent the E.164 number. The four HO-DSP octets of the private E.164 format do not have an E.164 meaning and do not have a preferred representation. Private networks that choose to use this format may base their own addressing on the E.164 address of the public UNI to which they are connected and use the DSP to identify ATM endpoints within their network.

The private E.164 format is useful for organizations that wish to use the existing largely geographically-based public ISDN/telephony numbering format. This AESA E.164 format can also be used by private endpoints setting up calls to endpoints in the public network.

4.3 IP Addressing

Internet Protocol (IP) addresses identify nodes in the Internet. Nodes can be hosts, routers, or any Internet-connected device that is addressable. Nodes in general are commonly known simply as *hosts*. IP addresses identify both a particular node and the network where the particular node resides in an internetwork. IP addresses are 32 bits (4 bytes) long and are expressed as four fields of one byte each, represented in *dotted decimal notation* (e.g, 18.227.0.25). [48]

The portions of the IP address corresponding to the network address and the node or host address depend on the type of address. The type of address can be found from the first byte of the address, as shown in Figure 4-4. Type A addresses have one byte of network address, leaving 3 bytes of node address. Type B addresses have 2 bytes of network address and 2 of node address. Type C addresses have 3 bytes of network address and only one byte left for the node address. Type A addresses are assigned to large networks with many hosts, while type C addresses are assigned to small networks with only a few hosts.

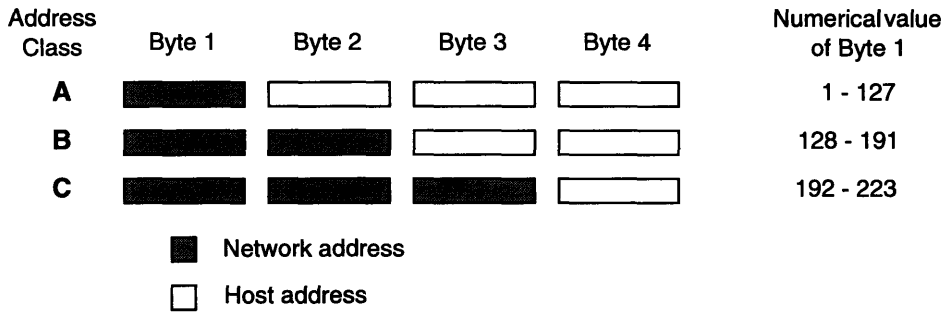


Figure 4-4. IP Address Classes

There also exist type D and E addresses. Type D addresses are used for multicasting to a group of hosts on a network. They have a value of 224 through 239 for their first byte. Type E addresses are reserved. They have the remaining values for the first byte, 240 through 255. Addresses of the form 127.x.x.x (class A) are reserved for the *loopback function*, provided for processes to communicate through TCP if they reside on the same host. These loopback packets are never sent out to the network.

The network portion of IP addresses for networks connected to the Internet is assigned through a central authority known as the Network Information Center (NIC). The host ID, or host address, is assigned by the local network administrator. If an IP network is never to interact with the Internet then even its network address may be assigned by the local network administrator, as the address will never have a chance to conflict with those assigned to networks in the Internet. However, it is usually the case that IP networks form part of the Internet.

Each individual site is usually assigned one network number. This network may then be subdivided into smaller subnetworks. These subnetworks may be physically separate or just logically separate. *Subnetting* is accomplished by trading a portion of the host address

space for subnetwork identifier space. An arbitrary number of bits at the beginning of the host address space can be used to identify a subnet. Normally all subnets within a network will use the same number of bits as their subnet identifier. For simplicity, multiples of one byte are normally used for subnetting.

IP hosts are configured with their IP address and the *netmask* they should use. The IP address is logically ANDed with the netmask (in a bit-wise fashion) to obtain the network (or subnetwork) address. For example, if a host has IP address 18.227.0.25 and netmask 255.255.255.0, then the network number is 18.227.0 and the host number is 25.

It is possible to assign more than one network number to the same network. This means that one network may have more than one network number on the same physical cable plant. This is sometimes done in order to allow class C networks to have more than the 254 hosts normally assignable to a class C network.

4.4 MAC Addressing

MAC addresses come from the Medium Access Control (MAC) layer of LANs, such as Ethernet or Token Ring. The data-link layer of a LAN is comprised of two sublayers: the MAC and Logical Link Control (LLC) sublayers (see Figure 4-5). The MAC layer is responsible for framing and addressing packets, as well as for error detection.

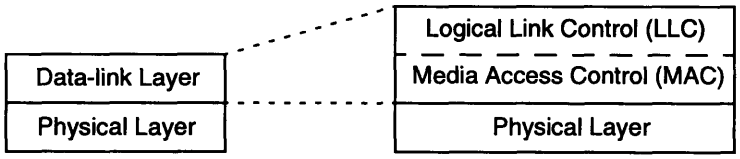


Figure 4-5. Data-link Layer of a LAN

MAC addresses are the data-link layer addresses in LANs, often referred to as physical layer addresses since they are used for each hop along the physical layer. MAC addresses identify the physical network interface of a host on a LAN. MAC addresses are universally unique addresses placed in each network interface card by the manufacturer. If a machine's network interface card is changed then its MAC address also changes. MAC addresses are usually 48 bits (6 bytes) long. They can also be 16 bits long, but this shorter length is very seldom used.

The MAC addressing space is flat. Each MAC address is much like a serial number. There is a central addressing authority (the IEEE) that ensures that no two interface cards will ever have the same MAC address. This addressing scheme allows for 2^{48} addresses, one per card. [48]

Although MAC addresses were defined for use in LANs and are still essential to LAN operation, they are often also assigned to network interface cards for other protocols. An ATM network interface card in a workstation may have a MAC address assigned to it by its manufacturer. Although this MAC address is not necessary for ATM (unless used in a private network whose addressing scheme includes the MAC address, i.e., in the ESI), it may be useful when emulating LAN protocols over ATM. These LAN protocols would make use of the MAC address of the ATM network interface card in emulating their MAC layer.

4.5 SS7 Signaling Point Code Addressing

SS7 networks identify their nodes by Signalling Point Codes (SPCs). SPCs are defined in GR-246 [1]. These point codes are 24 bits (3 bytes) long and consist of several fields, each

one byte long: the Network Identification Field, the Network Cluster Field, and the Cluster Member Field.

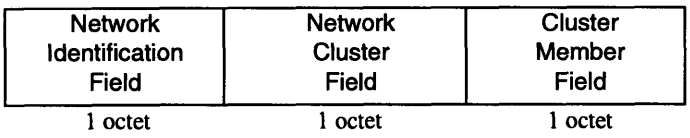


Figure 4-6. Format for U.S. Signalling Point Codes

The Network Identification Field identifies a network within the SS7 internetwork. Values for this field are assigned through a central authority.

Each network may be subdivided into clusters. The cluster in which a node resides is identified by the Network Cluster Field of the Signalling Point Code.

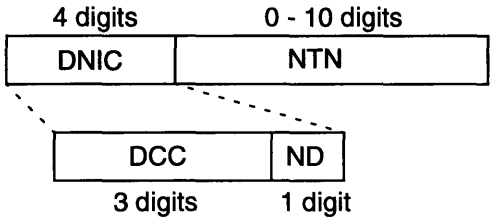
Finally, there is the Cluster Member Field, which uniquely identifies a node, or cluster member within a cluster of nodes.

Large networks are assigned a Network Identification Field code. The network administrator may then assign Cluster codes within the network. Small networks are assigned a Network Identification code and also a Network Cluster code. Thus one Network Identification code can be used for several small networks. The networks are distinguished by their Cluster Identification code. Very small networks are assigned a Network Identification code, a Network Cluster code, and a range of Cluster Member codes. Many very small SS7 networks thus share a Network Identification code and a Network Cluster code.

4.6 X.121: International Numbering Plan for Public Data Networks

ITU-T X.121 [19] defines the address structure used in X.25 networks. An X.121 address uniquely identifies an endpoint on an X.25 network by country, network, and end interface.

X.121 addresses consist of two parts, a Data Network Identification Code (DNIC) followed by a Network Terminal Number (NTN).



DNIC Data Network Identification Code
DCC Data Country Code
NT Network Digit
NTN Network Terminal Number

Figure 4-7. X.121 Address Format

The DNIC is comprised of a three digit Data Country Code (DCC) followed by a Network Digit (NOTE: these Data Country Codes should not be confused with DCCs of section 4.2.1 which are defined in ISO 3316, as they are different codes). The DCC identifies a country, while the Network Digit identifies a specific data network within that country. Up to 10 Public Data Networks (PDNs) may exist per DCC. Multiple DCCs may be assigned to a country with more than 10 PDNs. This system of DNICs can provide for 600 DCCs and a theoretical maximum of 6000 DNICs if all 10 Network Digits were allocated for each DCC.

The NTN can range from 0 to 10 digits. The format of the NTN is determined by the network provider identified by the DNIC.

5. Existing Address Translation Protocols

This section discusses protocols and methods of translating between different types of addresses in several common networks. Sections 5.1 through 5.4 discuss protocols that were not originally designed for use in ATM. These include ARP, the Address Resolution Protocol created for use in Ethernet LANs; DNS, the Domain Name System designed for name translations in the Internet; Intelligent Networks (INs) designed with telephony networks in mind; and E.164 – X.121 interworking which brings ISDN and X.25 networks together.

Instead of address translation, the purpose of RFC 1577, ATM Forum LAN Emulation, and the Next Hop Resolution Protocol (NHRP), which are discussed in sections 5.5 through 5.7, is to emulate LANs or IP networks on ATM networks. The great advantage of IP over ATM or LANs emulated over ATM is that existing applications that currently run only on IP networks or LANs can operate on ATM networks thanks to these protocols. These three protocols assume that the underlying ATM network is uniform, they do not account for differences in ATM addresses, for example, between public and private networks — the topic which this thesis addresses.

5.1 Address Resolution Protocol

In general each protocol layer can have its own addressing scheme. In a LAN, for example, MAC addressing is used at the data-link layer (layer 2) while IP (Internet Protocol) addressing may be used at the network layer (layer 3). The network layer uses IP addresses to get from end to end, often across several networks, while the MAC address is used to relay messages within one particular network. At each hop of the way a MAC addresses are needed to forward datagrams to the next hop towards the destination IP address.

The Address Resolution Protocol (ARP) [28] resides between the network (e.g., IP) and the data-link layers (LLC and MAC). It translates between the 32-bit IP address (network layer) and the 48-bit MAC address (data-link layer). Hosts running the Internet Protocol (IP) refer to each other by their IP addresses. When they send packets to each other they address them with their IP addresses. These packets are handed from higher layer protocols to IP, at the network layer; the IP layer adds its header with the relevant information and passes the data down to the data-link layer. At the IP (network) layer the packet is addressed with IP addresses. However, the data-link layer only understands MAC addresses. ARP is used to "translate" from the IP address to the corresponding MAC address that needs to be used for the next hop towards the destination IP endpoint.

If host *A* wishes to connect to host *B*, but host *A* only has the IP address of host *B* then host *A* broadcasts an ARP Request packet. This ARP packet contains *B*'s IP address and *A*'s IP and MAC addresses. The packet is broadcast specifically because *A* does not know *B*'s hardware (MAC) address and therefore cannot send any packets directly to *B*. However, although it is broadcast, only the host with that IP address, host *B*, will reply. *B* sends an ARP Reply back to *A* containing *B*'s IP and MAC addresses. *A* then enters *B*'s <IP, MAC> address pair in its ARP table. *A* can now send IP datagrams to *B* using its IP and correct MAC addresses.

The above example demonstrates how ARP allows a dynamic mapping between the addresses of two different layers, IP and MAC. ARP is not actually part of IP. It was introduced to allow IP to operate on LANs, where the physical media uses MAC rather than IP addresses.

This same ARP concept is used in LANs that are emulated over ATM, as is explained in section 5.6.

A related protocol is the Reverse Address Resolution Protocol (RARP) [29]. RARP is used when a network station knows its MAC address but does not know its IP address. This could happen, for example, in diskless workstations. The client machine sends a RARP Request to a RARP server somewhere on the network. The RARP server responds with that station's IP address.

As can be seen, there are several applications of this client/server address resolution system. Some of these are discussed below.

5.2 Domain Name System

The Domain Name System (DNS) is a hierarchically organized name service that allows users to establish connections to network stations using humanly readable names instead of cryptic network addresses [49]. Networks use network addresses in order to establish a connection between two network endpoints. However, humans would rather deal with names instead of network addresses. Users may use DNS to obtain a translation from an easy-to-remember name to the network address needed for routing. DNS was created originally for Internet mail and host address support. Its operation and protocols are defined in RFC 1034 [30] and RFC 1035 [31].

Figure 5-1 shows what part of the DNS tree structure may look like. Each node in the tree is labeled with a simple name (e.g., *edu*, or, *mit*, or *chimichurri*). The label can be up to 63 characters long. The root of the tree has a null label. The full domain name of a node is the sequence of labels from the node to the root, expressed with dots separating each domain name (e.g., *chimichurri.mit.edu*). All sibling nodes (nodes that are children of the same parent) must be named uniquely. This makes all full domain names unique. Note that this

allows nodes to have the same name as some of their descendants (e.g., *mit.mit.edu* is an allowable domain name).

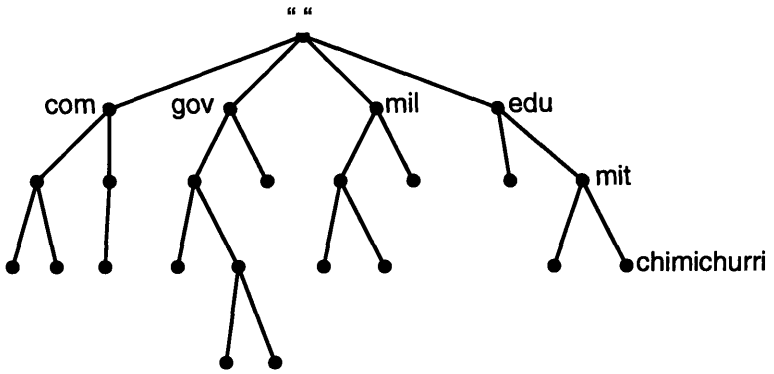


Figure 5-1. DNS Hierarchical Structure

Any subtree of the domain name space is a *domain*. For example, anything under *mit.edu* is in the *mit.edu* domain. Domains themselves are made up of domain names and other domains. A host also has its own domain — it is a domain (or subtree) made up of only one node (a leaf of the tree).

Domain names are just indexes into the DNS database. The domain names point to information about individual hosts. They can also point to information about the domain's children, or subdomains. Each host on a network has a domain name. Hosts in a domain are related logically, often by organizational affiliation or geography, and not necessarily by network or address. Hosts in the same domain may even be in different parts of the world and on different networks. Hosts may also have any number of *domain name aliases*.

The information associated with domain names is kept in Resource Records (RRs). RRs are divided into classes, the most popular class being the Internet class. RRs also come in

several different types. The types correspond to various different types of data that may be stored in the domain name space. The most commonly found type is the address RR. Other RRs store data such as mail routing information. RFC 1706 [36] defines the format of a new RR type that may contain NSAP addresses. These records could contain ATM addresses in AESA format, allowing a mapping from domain names to ATM addresses.

The top-level domains shown in Figure 5-1 (*edu*, *gov*, *mil*, and *com*) are only some of the existing top-level domains. There are also ISO 3166 top-level domains representing each country identified by ISO 3166. The *us* domain has 50 subdomains corresponding to the 50 states. For example, under the *us* domain there is a domain for Massachusetts named *ma.us*.

The programs that store information about the domain name space are called *name servers*. Name servers normally have complete information about a part of the domain name space, called a *zone*. The name server is said to have *authority* for that zone. Name servers can have authority for more than one zone. In fact, the root name servers happen to have authority for all of the top-level domains in the U.S. (*edu*, *com*, *gov*, etc.). The programs that are clients of and access name servers are called *resolvers*. A program running on a host that needs information about a domain uses the resolver.

Originally, when the Internet was small, there was a *hosts* file that all hosts had. This file contained the name and network address of all other hosts on the Internet so each host could resolve names to network addresses on its own. This method did not scale well as the Internet grew. Nowadays name servers keep track of host names (domain names) and their corresponding Internet addresses, along with other useful information. The naming authority is delegated to individual institutions on the Internet rather than having one central authority.

To obtain the network address of a host, e.g., *chimichurri.mit.edu*, a resolver queries the root server asking for the address of the *edu* name server [49]. The *edu* server is then queried to get the nameserver address for the subdomain *mit.edu*. Once the address of this last name server is obtained one last query to the *mit.edu* name server produces the network address of *chimichurri.mit.edu*.

For each domain there are several synchronized servers running together, so if one or even several name servers are down the name service for that domain can still be accessed. Since the root servers happen to be the servers for the top-level domains, such as *edu*, a single query to a root server produces the addresses of name servers for the *mit.edu* domain. Domain name clients cache recently used <name, address> mappings. This reduces the need to query the DNS every time an address is needed. The operation of the DNS is transparent to the user. For example, a user who wants to connect to a remote host simply gives the name of a host and the connecting application then queries DNS to get the corresponding address.

Inverse queries are also possible with DNS. A domain name may be found given a network address.

5.3 Intelligent Networks

A common use of Intelligent Networks (INs) is for number translation in telephony networks. "800 number" translations, for example, are carried out by Intelligent Networks. When a user dials an 800 number, a switch queries network databases to obtain a translation from the 800 number to the corresponding subscriber number that is to be used for routing the call to its destination. These network databases are a type of Service Control Point (SCP) known as a Line Information Database (LIDB). SCPs in the Advanced Intelligent

Network (AIN) also contain service logic to perform additional services for subscribers such as call forwarding that can depend on time of day or day of week.

TCAP (Transaction Capabilities Application Part) [1] is the database query protocol used in Intelligent Networks. TCAP is a part of the SS7 specification.

The same principles of Intelligent Networks that are used in telephony to access databases in telephony networks could be applied to Broadband networks as well. Some possibilities are discussed in section 8.

5.4 E.164 — X.121 Interworking in ISDN

The ITU-T has defined Recommendations to allow ISDNs to be interworked with X.25 networks. Two ways of interworking the two types of networks are:

1. using the dial-in method and
2. directly connecting the networks through a network-network interface.

Using the dial-in method, the user (or calling terminal) dials into the other network and then sets up a call within the other network. This is a two stage process. No dialing-in is necessary if both networks are directly connected through a network-network interface.

For address interworking, there are two methods commonly used, one method uses escape codes and the other uses a Numbering Plan Identifier (NPI). In the escape code method, an escape code digit is dialed or included with the address to indicate to the switches in the network that the address that follows the escape code belongs to a different addressing scheme. Note that escape codes must be uniquely defined within an addressing scheme so as not to be confused with a valid address in the domain of the addressing scheme.

The NPI method requires the use of a call control protocol and the existence of an NPI field within the protocol message that passes the called and calling addresses. The NPI field contains a code that indicates which numbering plan the called (or calling) address belongs to. Switches in the network must understand how to properly interpret the NPI field. The NPI method has the advantage of being non-ambiguous — the numbering plan is clearly and uniquely identified by the coding in the NPI field. It is not context dependent and it provides expansion capability for adding new addressing schemes.

ITU-T Recommendation X.31 describes support for X.25 equipment on an ISDN [17], and together with ITU-T Recommendation E.166/X.122 [12] support for interworking ISDNs and X.25 networks. X.31 contains recommendations for switched and non-switched (or permanent) connections. Since this thesis focuses on switched calls, we will only look at the recommendations for switched connections. X.31 talks about PSPDNs (Public Switched Data Networks) — X.25 networks are PSPDNs. The following paragraphs discuss how an X.25 DTE (Data Terminal Equipment) on an ISDN can connect with another DTE on an actual X.25 network. X.25 terminal equipment on an ISDN is referred to as a X.31 terminals because they abide by X.31 to operate on the ISDN.

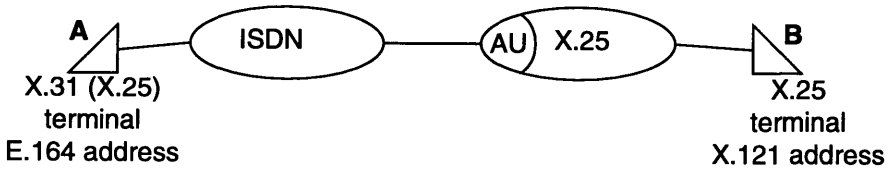


Figure 5-2. Circuit-Switched ISDN — X.25 Interworking

Figure 5-2 shows how an ISDN may be connected to an X.25 network so that an X.31 terminal in an ISDN and an X.25 terminal on an X.25 network can communicate. Circuit switched ISDN is used in this case, therefore the dial-in method is used. For *A* to set up a call to *B*, *A* first calls the AU (Access Unit) of the X.25 network using ISDN signalling. The called party address is the E.164 address of the AU. The AU looks like an X.31 terminal to the ISDN. The calling party address is the E.164 address of *A*. Once *A*'s ISDN connection to the AU is established, *A* requests a connection to *B* in the X.25 network using X.25 signalling. In the X.25 signalling messages, the called party address is the X.121 address of *B*, while the calling party address is still the E.164 address of *A*. If the escape code method is used, then addresses foreign to the network (i.e. an E.164 address in a signalling message in the X.25 network, or an X.121 address in a signalling message in the ISDN) are preceded by an escape code. If instead the NPI method is used, all addresses are accompanied by the appropriate NPI.

In this connection, the originating call is set up over the B-channel towards the X.25 network port (the AU) using ISDN signalling procedures. Then X.25 layer 2 and 3 functions are started and the originating terminal interacts directly with the X.25 network. Two separate numbers are used for access from the ISDN to the X.25 network: the ISDN number of the AU, and the address X.121 address of the called terminal.

For *B* to set up a call to *A*, *B* uses X.25 signalling to set up a call to *A*. The X.25 network sends the call to the AU. The called party address used is the E.164 address of *A*. The calling party address is the X.121 address of *B*. The AU then calls *A* using ISDN signalling. The called party address is the E.164 address of *A*. The calling party address used to set up the call from the AU to *A* is the E.164 address of the AU. Once the call from the AU to *A* is set up, the AU uses X.25 signalling to establish the X.25 call between *B* and *A*. The called

party address is still the E.164 address of *A*. However, the calling party address used in setting up the X.25 call is the X.121 address of *B*.

If packet switched ISDN is used instead of circuit switched, the ISDN can be connected directly to an X.25 network through X.75[1] or an equivalent protocol without having to go through an access port (or AU). This is shown in Figure 5-3.

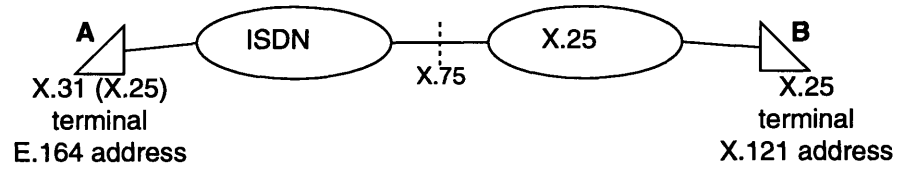


Figure 5-3. Packet-Switched ISDN — X.25 Interworking

In this case, an initial setup between *A* and the ISDN is needed, but from then on X.25 signalling can be used all the way from *A* to *B*. For a call from *A* to *B*, the called party address is always the X.121 address of *B*, and the calling party address is always the E.164 address of *A*, and vice versa for a call from *B* to *A*. The ISDN in this case may be assigned a DNIC (Data Network Identification Code) which X.25 uses for routing.

5.5 RFC 1577

The goal of RFC 1577 is to allow compatible and interoperable implementations for transmitting IP datagrams and ATM Address Resolution Protocol (ATMARP) requests and replies over ATM Adaptation Layer 5 (AAL5).

1. X.75 is the network-network interface used in X.25 networks [18].

RFC 1577 describes an initial application of "classical" IP and ARP in an ATM network configured as a Logical IP Subnetwork (LIS). The "classical" model refers to the treatment of the ATM host adapter as a networking interface to the IP protocol stack operating in a LAN-based (e.g., Ethernet or Token Ring) paradigm.

RFC 1577 assumes that private ATM networks use the private ATM address structure specified in the ATM Forum UNI specification, and public networks use either E.164 addressing or the private network ATM address structure.

In the Logical IP Subnetwork (LIS) scenario, each separate administrative entity operates and configures its hosts and routers within a closed LIS. Each LIS is independent of other LISs on the same ATM network. ATM-connected hosts communicate directly with other hosts within the same LIS, while communication with hosts outside of the local LIS is provided through an IP router. Routers can be members of several LISs. This means that a number of different, disjoint LISs may operate over the same ATM network. However, in order to keep with the LAN-based paradigm, hosts of different IP subnets must communicate via an intermediate IP router even thought it may be possible to open a direct VC (Virtual Circuit) between the two IP members over the ATM network.

The requirements for IP hosts and routers operating in an ATM LIS configuration are much the same as those imposed on LAN-based hosts and routers. All members of a LIS must have the same IP network/subnet number. They must use a router to access a host outside of the local LIS. They must have a mechanism for resolving IP addresses to ATM addresses via ATMARP and vice versa via InATMARP (Inverse ATMARP) when using SVCs. They must have a mechanism for resolving VCs to IP addresses via InATMARP when using PVCs. The ATMARP service has LIS scope only and serves all hosts in the LIS.

ATMARP is based on ARP (see section 5.1) with some extensions added to support ARP in a unicast server ATM environment. ARP was designed for use in networks with a shared communication medium, such as Ethernet where all stations broadcast over the shared medium, whether the packet is intended to be broadcast to all stations on the network or sent to only one host. ATM, on the other hand is connection-oriented. Connections must be set up between hosts before any data can be sent. Therefore the ARP method of broadcasting ARP Requests to all stations on a LAN would be very difficult to achieve over ATM. Instead, there is an ARP server. IP stations that wish to resolve an IP address to an ATM address send an ATMARP Request to the ATMARP server. This server in turn responds with the ATMARP Reply, if it knows the IP to ATM address mapping. Rather than the distributed architecture of the original ARP [28], ATMARP is centralized in one or a few servers. Each host implementation must know its own IP and ATM addresses. ATMARP supports E.164 addresses, AESAs, and it can also handle E.164 addresses with AESA subaddresses.

Each LIS must have a single ATMARP service. This may be offered either by a single server or multiple synchronized servers. This ATMARP service should be available to all members of the LIS who use SVCs.

RFC 1577 could be supplemented by the Next Hop Resolution Protocol (NHRP), which is described in the section 5.7. However, NHRP is not required of RFC 1577 hosts.

Members of a LIS (hosts and routers) have two lists. One is the ATMARP Request Address list, which contains one or more ATM addresses of ATMARP servers located within the LIS. The other is the NHS Request Address list, which contains one or more ATM addresses of Next Hop Resolution Protocol (NHRP) servers, also called Next Hop Servers (NHS). IP clients find the address of their ATMARP server and, if configured for NHRP, the address of their NHS from these lists.

IP clients reregister their ATM endpoint address with their ATMARP server using the address format of their ATM network (i.e., E.164 address or ATM private address). Server ATMARP table entries expire after 20 minutes. Client ATMARP table entries expire after 15 minutes. RFC 1577 specifies how nearby ATMARP servers should share address resolution information with each other.

RFC 1483 defines encapsulation methods for carrying connectionless network traffic over ATM networks[34]. RFC 1577 uses the methods of RFC 1483 to carry IP datagrams over ATM.

Appendix A of RFC 1577 is an informative appendix. It explains a method of embedding an IPv4 address inside an NSAP address using an automatic procedure within the ATMARP server. This would allow a LIS to operate using IPv4 addresses as ATM endpoint addresses without impacting client implementation. The entire process is carried out in a modified ATMARP server.

5.6 ATM Forum LAN Emulation

The ATM Forum's LAN Emulation (LANE) protocol is used to emulate a local area network (LAN) on top of an ATM network. The LANE specification [9] defines mechanisms for emulating either an IEEE 802.3 Ethernet or an 802.5 Token Ring LAN over ATM. By emulating a LAN, the LANE protocol offers higher layer protocols an interface which is identical to that of existing LANs. Thus, higher layer protocols do not need to be modified at all to operate over ATM. Existing higher layer protocols, such as IP or IPX, can be used and benefit from the high speeds of ATM.

There may be multiple emulated LANs operating in one ATM network. The LANE protocol resolves MAC addresses into ATM addresses. Using these address mappings LANE end systems can then set up direct connections between each other.

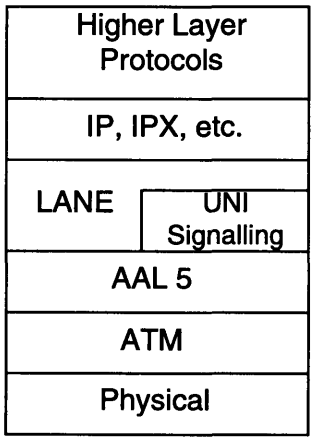


Figure 5-4. Protocol Stack of a LANE Host

There are several entities in the LAN Emulation environment. These are the LAN Emulation Configuration Server (LECS), the LAN Emulation Server (LES), the Broadcast and Unknown Server (BUS), and multiple LAN Emulation Clients (LECs).

The LEC performs data forwarding, address resolution, and other control functions for a single end system in an emulated LAN. It provides a standard LAN interface for higher layer protocols. Each LEC is identified by a unique ATM address and is associated with one or more MAC addresses reachable through that ATM address.

The LES implements the control function for a particular emulated LAN. There is only one LES per emulated LAN.

The LECS assigns individual LECs to particular emulated LANs by directing them to the LES that corresponds to the emulated LAN. There is one LECS per administrative domain. It serves all emulated LANs within that domain.

The BUS is a multicast server used to flood unknown destination address traffic, multicast, and broadcast traffic to clients within a particular emulated LAN. Each LEC is associated with only one BUS per emulated LAN. However, there may be more than one BUS in a particular emulated LAN.

Upon initialization, the LEC sets up a connection to the LECS. The LECS uses a configuration protocol to give the LEC the information it needs to connect to its emulated LAN, such as the ATM address of the LES, the type of LAN being emulated, etc. The LEC may then close the connection to the LECS. The LEC then sets up a connection to the LES and registers its own MAC and ATM addresses with the LES. If the LEC is a bridge it may also register other MAC addresses for which it is a proxy. A bridge can connect a legacy LAN to a LAN emulated over ATM.

During operation, a LEC may receive a packet to transmit from a higher layer protocol. The packet would be addressed to a MAC address. The LEC uses LAN Emulation ARP (LE_ARP) to resolve the MAC address to an ATM address. The LES acts as the LE_ARP server. While the LEC waits for the response to its LE_ARP request it forwards packets to the BUS. The packets are then broadcast throughout the emulated LAN. Once the LEC receives a LE_ARP response it sets up a connection to the destination LAN node and uses this connection to transfer data rather than broadcasting data through the BUS. If the LEC already has a connection open to the destination node then it may use it. LECs cache MAC to ATM address mappings for future reuse. The mappings are typically cached for about 5 minutes.

The LANE protocol allows existing protocols that operate on LANs to operate on LANs emulated over ATM. It also provides a method of interworking legacy LANs with emulated LANs such that higher layer protocols operating simultaneously over legacy and emulated LANs may never know they are on different media (e.g., Ethernet and ATM).

5.7 Next Hop Resolution Protocol (NHRP)

The classical model of LAN, as implemented by RFC 1577, forces hosts operating with IP over ATM to go through routers when they wish to communicate with hosts in different LISs, even if they may able to communicate directly via ATM. NHRP avoids this need to go through extra hops of routers when the source and destination hosts belong to different LISs. An NHRP server provides the source station with an inter-LIS address resolution mechanism so that if both stations are on the same ATM network, they can exchange packets without having to send them through a router. If the destination station is not part of the same ATM network then NHRP provides the source with the address of the egress point towards the destination. NHRP applies to Non Broadcast Multiple Access (NBMA) networks, such as ATM, rather than ATM specifically. It is, however, entirely applicable to ATM.

NHRP is an inter-LIS address resolution protocol, not a routing protocol. It provides the source with the NBMA (Non-Broadcast Multiple Access) address of the destination, if the destination is directly attached to the NBMA. If the destination station is not attached to the NBMA, then NHRP provides with the NBMA address of the exit router.

The Next Hop Resolution Protocol (NHRP) is currently in Internet Draft form [37]. It is an evolving protocol and should be recognized as work in progress. However, it is receiving

a great deal of attention from industry, and there may be ATM equipment that supports NHRP, despite its not having reached definite standard status.

The NHRP service provides an IP-to-ATM address resolution service for target hosts located outside the LIS. With NHRP a classical host, after not receiving a satisfactory reply from the ATMARP service, may then query the NHRP service for resolution. If the target IP address is known to the NHRP service an ATM endpoint address is returned for the query. The host may then open an ATM connection to that host. This configuration allows IP hosts to "cut through" the classical LIS boundaries on a case by case basis in favor of more direct ATM connections between cooperating IP members.

5.8 Summary

From the address translation protocols studied in this section we can see that there are many ways to translate addresses. The two types of addressing schemes identified are those in which addresses are organized hierarchically such as IP addresses and Internet domain names, and those with a flat address space such as MAC addresses.

Because MAC addresses identify hosts on a local shared-medium network, their address space is flat rather than hierarchically organized. Translations to obtain MAC addresses are done by broadcasting the translation request over the shared-medium network, as in ARP

Translations for IP addresses from Internet domain names (both addressed hierarchically) are done through queries to a database in a client/server fashion. The hierarchical nature of the addresses allows the database to be distributed, also hierarchically. In the case of DNS, several consecutive queries to different parts of the database are normally needed to get from the top level of the DNS tree to the desired node. The lookup is done by either the user or the user's application.

Intelligent Networks also offer client/server lookups in a database, such as for telephone number translations. The telephone numbering space also has a hierarchical structure. In Intelligent Networks the query to the database is done by the network on behalf of the user. Switches in the network look up the necessary translation. Neither the user nor the user's application have to do any part of the lookup.

In the ISDN–X.25 interworking example we saw how two different addressing schemes for network addresses (OSI layer 3 addresses) from different networks can be interworked. Two methods for interworking are the dial-in method which consists of two-stage dialing, and the "direct dial" method where both networks can signal to each other and can understand each other's addressing scheme.

RFC 1577, the ATM Forum's LAN Emulation protocol, and the Next Hop Resolution Protocol all take the idea of broadcasting address resolution requests (as ARP does in shared-medium networks) and apply it in networks that do not have a shared medium. The protocols achieve broadcasting by using a "broadcast" server. All hosts connect to the broadcast server and "broadcast" information through the server. The server resends broadcast messages to all hosts connected to it.

6. ATM Public and Private Address Translation

Large area, public Broadband networks will be deployed over the next few years. However, small private Broadband networks are already being deployed, often using addressing schemes that are not compatible with the E.164 addresses currently used for public networks. Whether interworking is a requirement or a value added service, it is desirable for all these networks to be capable of communicating with each other, even though they may use different addressing schemes.

Many existing ATM networks and some networks that will be deployed in the near future support only Permanent Virtual Connections (PVCs). PVCs are set up "by hand" through network management functions. Using PVCs, paths may be set up from end to end through different networks regardless of the addressing used in these networks. Some newer networks and most networks that will be deployed in the future will also support Switched Virtual Connections (SVCs). SVCs are set up from end to end by signalling. This is where endpoint addresses come into play. The following discussion on interworking public and private networks focuses on setting up SVCs across networks. Once an SVC is set up, switches route ATM cells using the VPI/VCI obtained during setup. In addition, only calls which transit the Public network are of interest. Calls that transit only Private networks are taken care of by P-NNI and are not of interest here.

Public and private ATM networks may be interconnected to create an internetwork like the one in Figure 6-1 below.

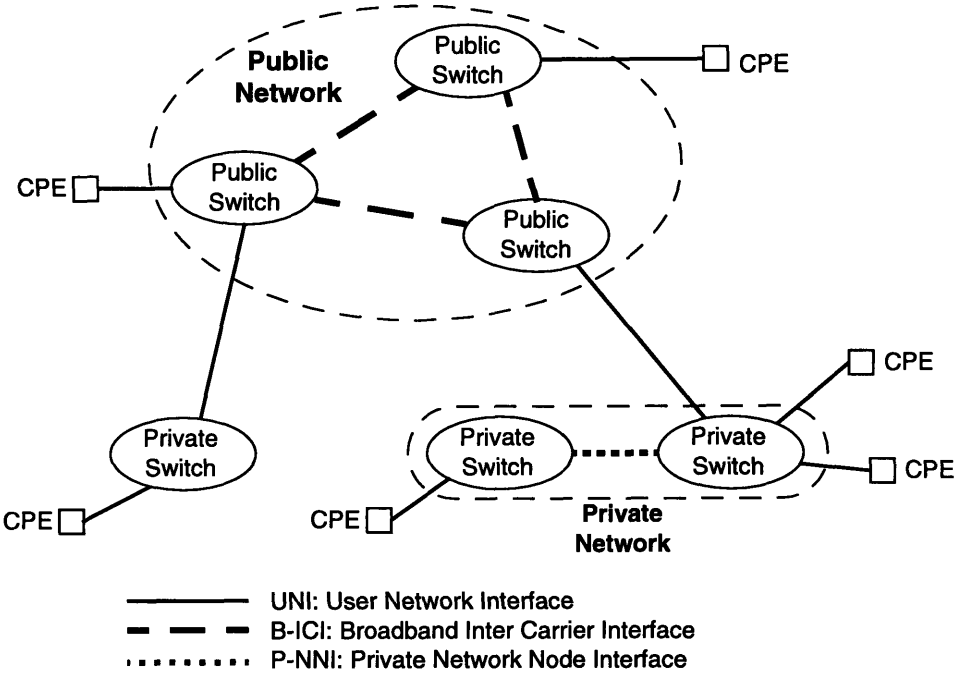


Figure 6-1. ATM Internetwork Interfaces

The interface between a user and a network is a UNI (User Network Interface). Interfaces between networks, or between switches in a network, are NNIs (Network to Network Interfaces, or Network Node Interfaces). The UNI has been defined by the ATM Forum [6]. The most recent UNI specification is version 3.1 (known as UNI 3.1). UNI 4.0, which will include support for multipoint-to-multipoint connections, is up for "straw ballot" by the ATM Forum in June, 1996. The ATM straw ballot means that the specification is "released for test vote with comments (may require 2 or more rounds)," and that a discussion phase follows.

The ATM Forum UNI specification is used uniformly in both public and private networks. UNIs in public networks are called Public UNIs, and UNIs in private networks are called

Private UNIs, but they follow the same ATM Forum UNI specification [6]. NNI specifications, on the other hand, are not uniform across networks. Public networks follow the specifications of the Broadband Inter Carrier Interface (B-ICI) [7]. Private networks, on the other hand, follow the Private Network-to-Network Interface (P-NNI) specifications [8], defined by the ATM Forum. Lastly, interfaces between public and private networks are UNIs rather than NNIs [6]. Private networks are treated as users of the public network, and as such their interfaces to the public network are UNIs. There has been some interest expressed in using an NNI interface, such as P-NNI between private and public networks. This may be an issue to look for in the future.

Note that B-ICI is the interface used between switches in a public network and also for interconnection of networks across administrative boundaries. The Broadband Inter Switching System Interface (B-ISSI) had started to be developed for intra-network connections (between switches in the same network), while B-ICI was being developed for inter-network connections. The B-ICI and B-ISSI efforts were combined to make what is now the evolving B-ICI. This differentiation between B-ISSI and B-ICI no longer exists, so interfaces between switches and between networks are both B-ICI, much like P-NNI is used in and between private ATM networks.

6.1 Network Interconnection Topologies

A call between two users can transit public and private networks in several combinations. Since users may be on either public or private networks, calls between them may transit both public and private networks in any order. Signalling along the path of the call will vary depending on the topology of the networks between the users — how they are interconnected.

A call originates in an ATM network when a SETUP message is sent across a UNI into the network. In the public network, the SETUP comes from either a user on the public network or from a private network (in which case the private network is a "user" of the public network). In order to connect this user to the requested called party, the public ATM network must figure out how to route the call. If the called party number in the SETUP message is either an E.164 address or an AESA E.164 address then the public network has enough information on which to base its routing decision. However, if the address is not E.164 then the network needs a method for determining how the call should be routed.

Consider the cases described in sections 6.1.1, 6.1.2, 6.1.3, and 6.1.4 with their respective figures where calls are set up between CPEs on public and private networks. These cases cover the four most common combinations of two users on public and private networks, for calls that transit the public network at some point. Note that the case where both users are on the same private network is not covered as this call never transits the public network. It is assumed that in general a call transits the public network only once. It is also assumed that if a call transits both public and private networks, the path traverses at most one private network before entering the public network, and at most one private network after exiting the public network. This means that the longest path considered is the path of a call that originates in a private network, transits the public network, and terminates in a second private network.

For convenience the public network is treated as having only two switches, and each private network is treated as having only one switch. Although this is a simplification it does not come with a loss of generality. Any signalling between two public switches can be extended to signalling through more switches, or simplified to include only one switch. Signalling in private networks can be extended to more than one switch by using P-NNI, the protocol used between private switches and between private networks. It may be

possible to extend the assumption of only one private network on either side of the public network to more than one network by using P-NNI.

6.1.1 Case 1: Public to Public

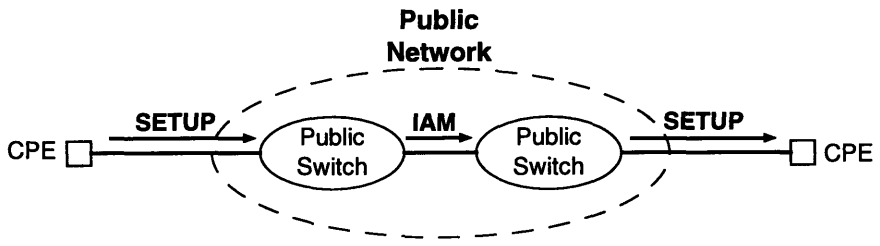


Figure 6-2. Case 1: Public to Public

In Case 1 (Figure 6-2) a call is set up between two CPEs that are on the public network. In this case the originating CPE sends a SETUP message to the public network. The SETUP message contains an E.164 address as the called party number, the address of the called CPE. The public network routes this call using the E.164 address as it would any other SVC call. The E.164 address is copied into the IAM (Initial Address Message) by the first switch and then into the final SETUP message by the last switch.

6.1.2 Case 2: Private to Public

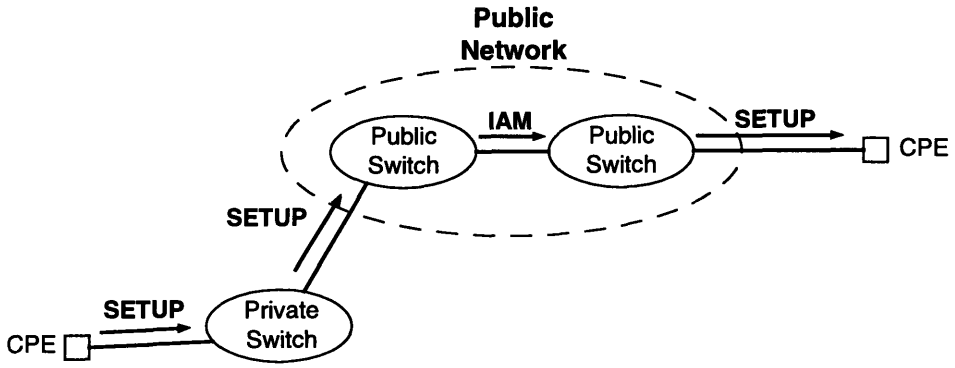


Figure 6-3. Case 2: Private to Public

In Case 2 (Figure 6-3) a call is set up from a CPE in a private network to a CPE in the public network. The called party number in the initial SETUP message is the E.164 address of the called CPE. Since the call is initiated in a private network, this address is in AESA format (see section 4.2.3). The private network would know to route any call to an E.164 address to the public network, unless the private network itself uses E.164 addressing. The private network sends a SETUP message to the public network over its public UNI. This SETUP message contains a called party number which may be in one of two forms: E.164 or AESA E.164. Both formats can be supported by the public network — the Public network can accept either an E.164 or AESA E.164 called party number [7].

If the private switch converts the AESA E.164 address into a regular E.164 address before sending the SETUP message, the public network then routes the call just as it did in Case 1 above. If the private switch does not perform any conversion, the Public network extracts the E.164 portion of the AESA E.164 for its routing.

Although the public network needs only the E.164 portion (the Initial Domain Identifier, or IDI) of the AESA E.164 address, the entire AESA (including the Domain Specific Part, or DSP) can be sent across the network to the called party CPE. In order to support certain higher layer protocols at the endpoints, the public network should transport the entire AESA from end to end. The public network may, however, charge an extra fee for including the full AESA in its end-to-end signalling.

The DSP (the remainder of the AESA after the E.164 part) is used by some higher protocol layers at the endpoints, such as LAN Emulation protocols (see section 5.6). Annex A of UNI 3.1, *Guidelines for Use of ATM Address Formats*, states that the originating CPE should set the DSP of the AESA E.164 address to zero if it is not used, indicating that no information is carried by the DSP [6]. The public network could tell whether or not it should carry the entire AESA to the destination CPE by looking for this indication.

Note that since the called party in this case is in the public network, its address will be E.164 and not an ICD or DCC AESAs.

6.1.3 Case 3: Private to Private Across Public Network

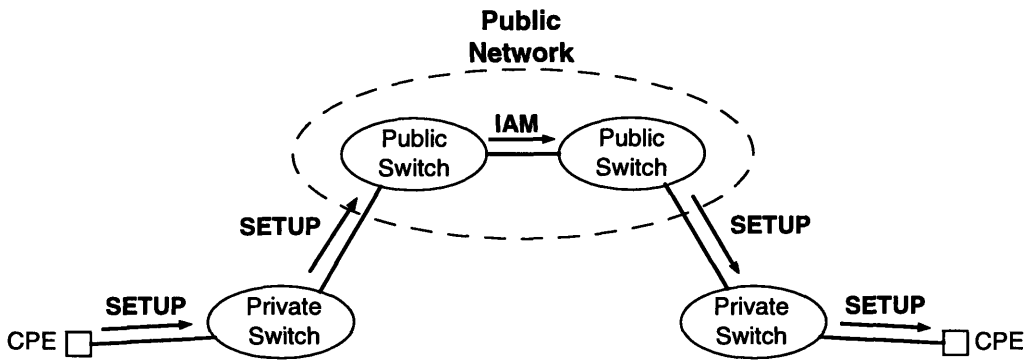


Figure 6-4. Case 3: Private to Private Across Public Network

In Case 3 (Figure 6-4) a CPE in a private network sets up a call to another CPE in a different private network. The two private networks do not have a direct interface between them but they are both connected to the public network, so the call is routed through the public network.

The originating CPE sends a SETUP message to its private network. The called party number in this SETUP message is the AESA of the destination CPE. The first private network recognizes that this private address corresponds to another private network. It therefore routes the call towards the public network in order to reach the destination private network.

The called party number may be in any of the three AESA formats (AESA E.164, ICD, or DCC). If it is AESA E.164, the call can be treated like the one in Case 2 and the entire address is carried through the public network because it will likely be needed once the call reaches the destination private network[1].

If instead the address is ICD or DCC then the destination private network's E.164 address (the address that identifies its interface to the public network — its UNI to the public network) must be obtained in order to route the call in the public network. The called party AESA must be sent through the public network to the private network.

ATM signalling messages allow carrying both an address and a subaddress. This capability was defined to allow "tunneling" through public networks. The translation from the called AESA to the destination private network's E.164 address can be done either by the private network before sending the SETUP message or by the public network after it receives the SETUP message. In either case, once the translation has been done, the called AESA is carried transparently through the public network in the Called Party Subaddress of an IAM. The E.164 address is carried in the Called Party Number of the IAM. Routing is done on the E.164 address in the Called Party Number parameter.

Another way of carrying the AESA information across the network was recently made possible by two new B-ISUP parameters that were recently added in the ITU. These are the AESA for Called Party and AESA for Calling Party parameters. These parameters are described in section 2.8.2.

These new parameters are used as follows. The destination private network's E.164 address is placed in the Called Party Number parameter of the IAM. This Called Party Number is used for routing. The AESA received from the user (the destination private address) is placed in the AESA for Called Party parameter. The AESA for Called Party parameter is then transported through the public network transparently.

1. If the destination private network uses pure E.164 (not AESA E.164) addresses (a very unlikely scenario) then the entire AESA need not be carried through to the destination private network. If, on the other hand, the destination private network makes use of the entire AESA for its addressing (a much more likely scenario) then the entire AESA must be carried through the public network.

Note however that the ITU currently allows the AESA for Called Party parameter for E.164 AESAs. This method would work if the destination private network's addressing scheme is E.164 AESA. However, calls to networks that use ICD or DCC AESAs would not be supported. Note also that in the case of a call to a private network that uses E.164 AESAs a translation may not be necessary. The calling user would request that the network set up a connection to an E.164 AESA; the network can then extract the E.164 information from the AESA and route on that address, assuming that the private network follows the same NANP guidelines that public networks do.

If the Called Party Subaddress parameter of an IAM is used to carry the called AESA through the public network then either the last public switch moves the AESA back into the Called Party Number of the SETUP message it sends the private network, or it could simply send the SETUP message with the E.164 number and the AESA in the subaddress. In the second case the private switch simply moves the AESA back into the Called Party Number before sending the call to the called party.

If instead the AESA for Called Party parameter of the IAM is used, the terminating switch in the public network takes the called AESA from the AESA for Called Party parameter and places it back into the Called Party Number IE (Information Element) of the SETUP message it sends to the private network. This functionality is explained in section 2.8.2.

The private network then routes the call based on the called AESA which is once again the Called Party Number.

These translations between AESA and E.164 addresses could be done on either side of the UNIs. Section 6.2 analyzes pros and cons of translating at either side.

6.1.4 Case 4: Public to Private

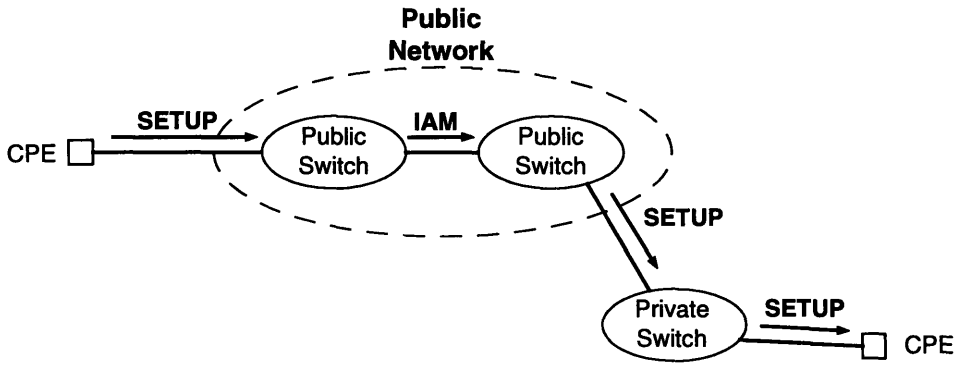


Figure 6-5. Case 4: Public to Private

In Case 4 (Figure 6-5) the originating CPE sends a SETUP message to the public network. This SETUP message contains either the destination CPE's AESA, in which case the public network will have to obtain the E.164 address of the destination private network, or it contains the E.164 address of the private network and the destination CPE's AESA as the subaddress. The call progresses much in the same way as in Case 3 above. Using either the Called Party Subaddress or the AESA for Called Party parameter of an IAM the called AESA is carried across the public network, as described in Case 3. The E.164 address of the destination private network is used for routing in the public network.

Once again, the second translation, which only really involves putting the called AESA back into the Called Party Number IE, is done either in the public network or in the private network and a SETUP message is sent from the public to the private network. The private network then routes the call to its destination using the destination CPE's AESA.

As in the previous case, the translations could take place on either side of the public UNIs, either inside the public network or outside of the public network. Outside the public

network means inside the private network in cases 2 and 3 but in case 4 the user would be responsible for the translation.

6.2 Options For Location of Address Translation

For the public network to route a call to its destination, it must have the E.164 address of an endpoint in the public network. The cases of section 6.1 showed how, upon entering the public network, the called address must be translated to an E.164 address if it is not already an E.164 address. The public network then uses this E.164 address to route the call within the public network.

This translation, from private to public, can be done either before or right after entering the public network. In order to perform the translation, knowledge about the point of attachment of the destination private network (the point where the call exits the public network and enters the destination private network) is necessary. This point in the public network is identified by the private network's address in the public network (an E.164 address).

Upon exiting the public network it may be necessary to replace the E.164 address with the original called party address, as described in the cases of section 6.1. Although this may be thought of as a translation, because the E.164 address is replaced with an AESA, it is very easy to carry out. The AESA is already carried through the public network, it simply has to be moved from one parameter to another. The switching of the AESA to the called party parameter can be done either inside the public network or outside, right after the call exits the public network.

In Case 2 the first translation is quite simple: an AESA E.164 address is converted to a regular E.164 address. The E.164 number can simply be extracted from the AESA E.164

address. The first translation in Cases 3 and 4 may be more complicated if ICD or DCC AESAs are used, the destination private network's E.164 address must be found.

The ATM address translation can be done at any one of several places in the internetwork. The translation can be done inside the public network, inside the private network, and it can also be done by the user. If it is done in the public network, users on the public network can take advantage of the translation service. Users on private networks can also take advantage of the translation service as long as their private network knows to route calls through the public network if the called AESAs is not an address in its own private network.

If the address translation is done in the private networks, users in these private networks will be able to use the translation service. However, it would not be available for users connected directly to the public network. These users would have to obtain translations in some other way, for example, through directory services.

One last location where the translation can be done is at the users' end. Users would be responsible for getting any translations they need. They could do so through, for example, directory services.

From the public network point of view there are two options to look at: translating inside the public network, and translating outside of the public network. The option of translating outside of the public network can be further broken down into translating in the private network, or translation by the user.

The following is an evaluation of the alternatives for the translation done upon entering the public network: translating in the public network, in the private network, or at the user's end. The alternatives are analyzed with respect to certain criteria.

6.2.1 Metrics for Comparing Options

We will use the following criteria to compare the options for where the address translation should take place.

- Flexibility: Ability to offer address translation service to any user.
- Growability/Scalability: Ability to handle increased demand for address translation service by adding additional resources.
- Operability: Ability for the system to be configured, maintained, monitored, and controlled.
- Deployability: Ability to be introduced and used in the context of the existing network and operations environments.
- Cost: The cost of the total resources needed for hardware, software, etc.
- Ease of Use: How easy it is for customers of the address translation service to use this service.
- Security: How secure is information in the system and information about customers.

6.2.2 Translation in the Public Network

The following analysis of pros and cons of translating in the public network is done with regard the criteria stated above.

- Flexibility: If address translation is done in the public network, the address translation service can be offered to any user or private network that is connected to the public network. If configured on a general basis, as 800 service is in telephony, the service can be offered to anyone connected to the public network. If configured on an

individual subscriber basis, the service could be customized for each user or for an entire private network.

- Growability/Scalability: Once a system is in place, the public network would cope with the addition of multiple private networks by expanding its databases. Public networks have shown an ability to handle large databases (e.g., 800 service and LIDB, the Line Information Database). Public networks can also currently handle a large volume of translations (again, as shown by their ability to handle 800 number calls).
- Operability: When a private network is connected to the public network it is assigned an E.164 address. Routing tables in the public network are modified accordingly. This is done for new private networks and end users alike. Public networks reconfigure their customer databases and routing tables whenever a private network connection is added, changed, or removed. With adequate operations systems in place it should also be relatively straightforward for public networks to update their address translation databases.
- Deployability: Depending on how the address translation service is implemented, much of the existing hardware and software, and knowledge of these systems used in public telephony networks for similar services, such as 800 number translation, may be reusable.
- Cost: The cost of implementing the address translation service will depend on the system used. If existing translation systems are re-used, the cost can be kept down. In any case, software in ATM switches will have to be added for the translation service, and hardware may have to be added as well for the switches to communicate with an address translation database. The entire cost would be in the public network.
- Ease of Use: Customers (both end users and private networks connected to the public network) would be able to make use of the translation service easily, since the entire

system would be inside the public network. When a user sets up a call to an AESA, the public network would take care of the translation.

- Security: Availability of private networks' E.164 addresses can be controlled by the public network much like telephone numbers can be unlisted. Also, access to a given private network may be controlled by making the E.164 number available only to certain users, or only certain interfaces can be made "public" by giving out only the E.164 addresses that correspond to the "public" interfaces.

6.2.3 Translation in the Private Network

- Flexibility: If the address translation is carried out in the originating private network, users of the private network would be able to make use of the address translation service to establish calls across the public network. The private network would need a way to obtain the necessary data for the translation table. The data would be distributed in the private networks rather than centralized as in the case of translation in the public network. Although address translation service can be offered to users of the private network, users connected directly to the public network would not be able to make use of the service.
- Growability/Scalability: Private networks would need a way of getting the E.164 addresses of new private networks as they are connected to the public network. If the public network helps out by keeping a *master* database which private networks can access as needed, the system should be growable. Private networks, especially small ones, are unlikely to be able to store the entire address translation database as more and more private networks are connected to the public ATM network. A system such as DNS, however, where several entities (possibly the private networks) store parts of the database might provide a more scalable solution, since the database can be distributed.

- Operability: In this case, private networks are responsible for getting address translations that are required by users. Automatic configuration and updating of the address translation database could be possible provided the private network can get the address translation information from somewhere. This seems to suggest, once again, that if the private network is to be responsible for the translation then a system should be used where the database is distributed.
- Cost: Private networks would have to cover the cost of implementing and maintaining the address translation service. The actual cost will depend on the system used. If private networks are to store translations that may be used by anyone on that private network, storage may be costly. However, if the database is distributed, such that the private network need only cache and keep track of certain translations, the storage problem is simplified.
- Ease of Use: As in the previous case (translation in the public network) this case makes it easy on the user. The user simply sets up a call to the desired AESA-addressed end point and the private network takes care of the translations. In this case, users rely on the private network for address translations. Note however that if the address translation service is offered in the private network, users connected directly to the public network will have to fend for themselves.
- Security: Availability of private network E.164 addresses may still be controlled. However, if the translation databases reside in the public network it may not be as easy to ensure that a an E.164 number is removed from the database if, for example, at some point in time the number is changed to "unlisted." If, on the other hand, a distributed database is used, then it would be easier for the keeper of a portion of the database to include or exclude entries in the translation database. If a private network does the address translation it can control outgoing calls if it so desires.

6.2.4 Translation by the User

- Flexibility: By making users responsible for address translation, any user capable of getting the appropriate translation can set up a call across the public network to an AESA-addressed endpoint. A system would be needed to get the translations to the user. This could be feasible with a system such as DNS where the public network, and possibly private networks, would actually have some involvement. For example, the public network would contain a translation database which the user would access. If we take into account a normal storage capacity in user equipment, only a limited number of translations could be stored at once. This suggests that however this system is implemented, it would only be feasible for the user's system to contain a small portion of all possible translations, possibly cached. The flexibility discussed here assumes some help from the network to which the user is connected. Without help from the network it would be much more difficult for the user to obtain translation information, unless the user only ever needs a few limited translations.
- Growability/Scalability: As discussed above, users (or their CPE) are very unlikely to be able to store all the translations they may ever need, unless they only ever set up calls to a few specific parties. Again it seems the only way to make the method of "translation by the user" scale is if users or their CPE access a database in the public network, for example. Users on private networks might access a database on their private network instead.
- Operability: In this case the user is responsible for obtaining the right ATM address translation. A system where the person must find the translation and enter it manually would be a tedious one, especially if the user wishes to set up connections to many different endpoints. An automated system would be much more easily operable. Such a system might query a database somewhere in the network for a translation when needed. This again brings the networks (either public and/or private) back into the

picture. The user would then have to ensure that his end of the translation mechanism is running correctly.

- Cost: Here again, the system used will determine the cost. Several factors to consider are how the system obtains the translations and where the address translations are stored. Are they stored by the user or by a database somewhere that the user can access?

- Ease of Use: If the user is responsible for address translations, ease of use will depend on the system used. A system where the user has to play an active role in obtaining the translations is likely not to be easy to use. On the other hand, a system where the user's CPE automatically looks up the translation in a database will likely be much easier to use. The user or site administrator will, however, have to maintain the system — a role not needed if the translation takes place in either the public or private networks.

- Security: As in the case of translation in the private network, availability of information would depend on where the database is. Availability of addresses could be handled by whomever gives out the information in the first place (i.e., either the public or the private network).

6.2.5 Summary

Translating outside of the public network implies a translation either by private networks or by the user. Translation by private networks forces users directly connected to the public network to translate for themselves. Therefore, ATM address translation should be considered done either inside or outside of the public network.

If done inside the public network it can be done transparently for all users, whether they are customers of public or private networks. If done outside of the public network there would

have to be either a method for all customers of the public network or possibly two different methods, one for private networks and another for users connected directly to the public network.

Instead of a translation service offered to users, a "look-up" system might be used. This approach could be modeled after the Domain Name System (DNS) of the Internet. A DNS-like approach can be implemented in many ways. One way is for private networks to register their addresses with an organization, possibly paying a fee as is done in the Internet. This organization would have authority for the registered private network and would provide translations to the rest of the internetwork. The only full address users would be required to keep is that of a domain name server. Another possible DNS-like implementation could resolve network names to E.164 addresses and end user names to AESAs. This would allow a two-stage approach much like the one used with POTS telephone numbers and extensions.

A DNS-like approach opens up many questions, such as who runs the service, who owns it and who pays for it. A DNS-like approach would likely share the translation responsibility between either the public or private network and the user. The Internet has demonstrated the usefulness of DNS. One question that would have to be answered if a DNS-like system is implemented in the ATM networks discussed it who has naming authority. The authority might lie in either the public network, private networks, or completely separate entities.

Another possible approach is through directory services. The user would access the directory service (like calling 411 on the telephone) and request the translation. This service could be provided by either the public or private network.

Translation in the public network has many advantages for its customers, and depending on the implementation it can provide a transparent translation service for its users. It can allow

users anywhere on an ATM internetwork obtain the necessary translations to communicate with each other. Complexities of transiting the public network to get to another private network are kept inside the public network and away from users.

In this thesis we choose to explore translation inside the public network, realizing that it is not the only option but it is reasonable and advantageous to both the users and the common carriers.

7. Options for Address Translation in the Public Network

In order to explore ATM address translation in the public network, four options for performing the translations are described in this section.

A switch-based method is discussed, where address translation tables reside in each switch. A method using X.500 directory services is also discussed. This method would require switches to be able to interact with the X.500 database. Another option is using a Domain Name System (DNS) like the one used in the Internet for address and name translations. Finally, a use of Intelligent Networks is discussed to see how INs, which are currently used in telephony networks, could be used for ATM address translations to facilitate interworking public and private ATM networks.

7.1 Switch Based

In a switch-based method of ATM address translation, end office switches contain ATM address translation tables. Only end offices need the tables because once the address is translated the call goes through the public network with the E.164 address obtained from the translation.

These tables would need to be kept updated, a task which could be quite difficult in anything other than a small network. Assuming each private ATM address is associated with one public ATM address (the public network address of its private network) there could be as many entries in the address translation table as there are private ATM addresses. It is possible for the table to be smaller if a clear hierarchical addressing structure is observed by private networks. If addressing is strictly hierarchical, then domains of addresses can be kept in the tables rather than all private addresses. It is not known at this time whether or not private ATM networks will follow a strict hierarchical addressing. But

even if they do, different private networks may use different hierarchies in their addresses. Recall from section 4.2 that there may be any number of ways in which the DSP of an AESA is subdivided.

Updates would have to be propagated to all switches at the edge of the network (end offices). AESAs or entire private networks can be made accessible or inaccessible from certain areas in the public network by making their translation available in the switch-based translation tables or by removing the entry from the table.

A switch-based solution may seem at first glance to be similar to the ISDN—X.25 interworking problem discussed in section 5.4. In some respects it is a similar problem. ISDN and X.25 networks are different networks and they must be able to reach each other in order to set up calls across the two networks. However, in the case described in section 5.4 where access to the X.25 network is through an Access Unit, the user has to know the E.164 address of the Access Unit so as to set up a "two stage" call, first to the Access Unit and then over X.25 to the destination.

In the other case described in section 5.4 the ISDN and X.25 networks share an X.75 interface. This lets users of packet mode ISDN specify a destination X.121 address. The ISDN then routes the call to the X.25 network as appropriate. This solution, in which both networks share a network-network interface (X.75 in this case) is not yet available for ATM. Current plans, as discussed in section 6, are for public networks to use B-ICI and for private networks to use P-NNI as their network-network protocol. The interface between private and public networks is still UNI. There has been some discussion in the ATM Forum about the possibility of extending P-NNI to interface private and public networks, but there is no effort in this area as of yet.

7.2 X.500

A possible alternative for deploying an ATM address translation service is using X.500. X.500 is the ITU-T standard for a sophisticated directory services system. It is an application-layer protocol in the OSI architecture [21,46]. Being a directory service it supports name-to-address and address-to-address translations.

In small network environments, directory services are typically centralized, with all the information residing on a single server. As networks grow, the directory service can be distributed to improve performance and availability of information. X.500 directory systems consist of three main functional components: the Directory Information Base (DIB), Directory System Agents (DSAs), and Directory User Agents (DUAs). The DIB is all the information maintained by the database. This information is stored in and managed by DSAs, the network servers. DSAs provide the actual directory service and implement the service side of the directory operations. DUAs represent the client side of the directory service, accessing directory information for the user.

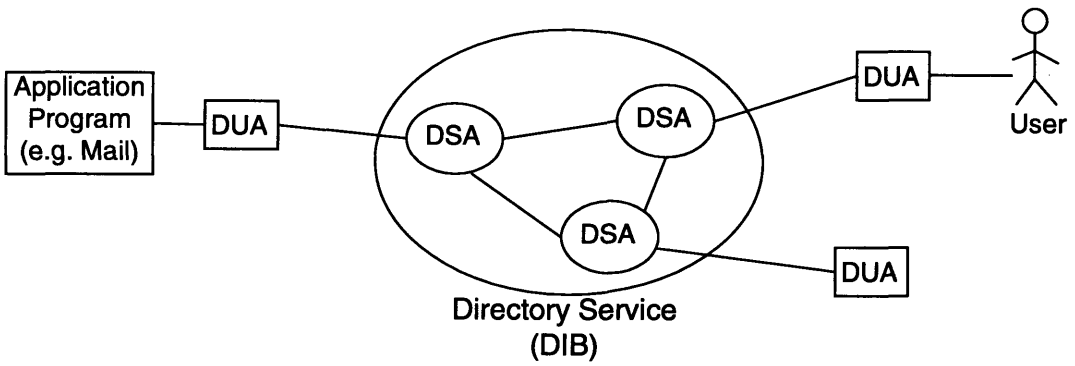


Figure 7-1. X.500 Distributed Directory Service

While the *physical* representation of the directory is the DIB, the *logical* representation of the database is known as the Directory Information Tree (DIT). Each DIB entry corresponds to a node in the DIT. Each directory entry is made up of attributes that provide specific information about particular characteristics of a network entity. An *attribute* consists of a type and a set of possible values of a specified syntax.

For AESA-to-E.164 translations using X.500 directory services, AESAs or groups of AESAs can be set up hierarchically in an X.500 directory tree. Each AESA or group of AESAs (e.g., grouped by the private network to which they belong) would have as an attribute the private network's E.164 address. A look-up in the directory would produce the E.164 address of the AESA's private network.

X.500 directory services are normally accessed by users or their applications. This suggests that a user or the users application would look up the translation before setting up a call across the public network. However, if Directory User Agent (DUA) capabilities are incorporated into switches, the switches could look up translations.

7.3 Domain Name System

Another option for ATM address translation in the public network is to use an Internet-like Domain Name System (DNS). DNS provides a client/server method of obtaining address translations. It is normally used to resolve Internet host names to IP addresses, but some of its Resource Records could be used for ATM address translations. In fact, there is a Resource Record that has been defined to contain NSAP addresses — AESAs are NSAP addresses. A more detailed description of DNS can be found in section 5.2.

In the Internet, users' applications are the most common clients of DNS. Before e-mail is sent across the Internet, or before any remote connection is set up, the application queries

DNS to obtain a translation from a host name to an IP address. The IP address is then used to establish the connection across the network. This again seems to suggest that if DNS is used for ATM address translations, users' applications would query DNS to obtain an the E.164 address that is to be used to reach a given AESA. Another possibility, however, is for switches, or nodes in the network, to be able to query DNS to obtain translations.

A DNS-like system could offer translations from AESAs to E.164 addresses, or from network names to E.164 addresses. It might also be used to provide translations from end user names to either AESAs or E.164 addresses. The system can be distributed within the public network so that different parts of the public network have authority over different domains, or it might be distributed such that the public network has authority over the top level domain and subdomains are handled by non-public entities.

Using DNS may even aid in the integration of public and private ATM networks with the Internet someday, since DNS is such an important part of the Internet.

7.4 Intelligent Networks

Intelligent Networks (INs) were described in section 3. Intelligent Network capabilities of public telephony networks are currently used for address translations, among other services, in narrowband telephony networks. This includes services such as 800 number translations. When a user dials a telephone number in which the area code is 800, the originating switch in the public network queries an SCP in the network. The SCP responds telling the switch to route the call to the real phone number that the 800 number maps to.

A similar service for AESA-to-E.164 ATM address translations would consist of a public switch querying an SCP when it receives a call setup to an AESA. The SCP would look up the translation and respond to the switch, telling it to route the call through the public

network to the destination. The destination would likely be the UNI of a private network that is connected to the public network.

Using IN services would allow public networks to reuse a great deal of their investment in INs for telephony. Some of the IN network components may need to be changed to support broadband data rates and setup times, but the basic network is already in place. Using INs for broadband can also help in interworking narrowband (telephony) and broadband networks, since public narrowband networks in the U.S. are Intelligent Networks. In addition, INs can offer a great deal of flexibility to ATM address translations. For example, a translation can be made available only to certain users in order to limit access to a private network, a translation can be changed depending on criteria such as time of day or day of the week, or even location of calling party. Much of this flexibility is offered by narrowband IN services today.

7.5 Conclusion

There are many options for translation in the public network. A switch-based implementation may be feasible initially while the network is small, and might remain feasible if AESAs in private networks are organized in a clean, hierarchical fashion. Switch-based translation does mean, however, that the public network would "understand" AESAs as well as E.164 addresses. This is not consistent with the public networks' current stance on only supporting E.164 addresses.

A directory services approach, such as DNS or X.500 is another option. Users or their applications, or possibly switches themselves would look up translations when needed. Because of the distributed nature of these services, they can scale well as networks grow and as new private networks are connected to the public network. One question to be

answered here is who would run the directory service. The public network could do so because it can get access to all the necessary information. It is also possible for an independent entity to offer the service.

Finally, Intelligent Networks, used in telephony for similar address translation services can be used. In this option, the whole ATM address translation system is contained within the public network. All the work of getting the translation is taken away from the user and left up to the public network alone. This IN-based service can offer users of the public network a transparent address translation service. Users would only have to know the AESA of the destination the wish to reach. The translation and routing would be taken care of by the public network.

The rest of this thesis explores how ATM address translation can be offered to users, modelling public ATM networks as Intelligent Networks.

8. Intelligent Network Based Address Translation

Section 6 discussed the possibilities of where ATM address translation may take place when a call transits the public network. Given the conclusions of that section, that translation in the public network would be reasonable and advantageous, this section goes into a method of translating ATM addresses in the public network using Intelligent Network based services, as discussed in section 7.

Public telephone networks in the U.S. are Intelligent Networks. They have all the elements and functionality of Intelligent Networks (INs) that were described in section 3. The same organizations that currently form the public telephone network, Local Exchange Carriers (LECs) and Inter Exchange Carriers (IECs), are also building public broadband networks. Therefore many of the Intelligent Network elements and services can likely be reused in the newer public broadband networks. The infrastructure for Intelligent Network services is, for the most part, in place. Using INs would allow reuse of a great deal of what the LECs and IECs have already invested in for Intelligent Networking. Using Intelligent Network services for these ATM address translations can allow public and private networks to interact, as described below.

Modelling public ATM networks as Intelligent Networks, the rest of this section investigates how ATM address translations can be carried out by Intelligent Networks, thus allowing calls to transit the public network, regardless of their origin or destination.

8.1 ATM Address Translations

ATM private-public address translations can be implemented as a service of Intelligent Networks. Refer to section 3 for a brief overview of Intelligent Networks. This section explores how the address translation service could be implemented in an IN.

The basic function would be as follows. The public network receives a SETUP message from an end user or from a private network. The SETUP message contains an AESA. If the AESA is in E.164 format the public network can route the call as it would any other call in the public network. However, if the E.164 address corresponds to a portable number, a database query may be needed. If the AESA is either ICD or DCC, then the public network queries a database which returns the E.164 address needed to reach the given AESA. The public network then uses this E.164 address to set up the call through the public network to its destination.

Recall from section 3.6 that standards for broadband IN are not yet defined, but current proposals are very closely modeled after narrowband IN. Differences from narrowband IN are mainly to account for inherent differences between broadband and narrowband networks and services, such as different addresses and different signalling protocols.

A possible deployment architecture for a Broadband Intelligent Network is one where all switches at the edge of the public network, those that interact directly with users, are IN-capable. These Broadband Switching Systems (BSS) would communicate with an SCP, or SCP-like device, for additional services such as address translations. Recall from section 6 that interfaces between switches in the public network are B-ICI, and interfaces between users and networks are UNI. The one additional interface that is introduced with IN is the TCAP interface between BSSs and SCPs.

BSSs and SCPs need not have direct physical links between them. Although IN-capable switches may be directly connected to SCPs, they may otherwise be connected through an STP or through other switches. The procedure for translating AESAs to E.164 addresses would be the same in all of the possible architectures, as long as the switches at the edge of the network have IN functionality.

A typical call through the public network with an AESA-to-E.164 translation is shown in Figure 8-1 and described below. Note that the logical interfaces between switches and the SCP are shown. The physical connection may not be direct, as discussed above. CPE1 shown in Figure 8-1 could be either an end user or a private network. Both an originating end user and an originating private network are treated generically as CPE in this section in order to focus the discussion on procedures in the public network without worrying about the details of the originating private network. BSS1 and BSS2 are in the public network. CPE2 is connected to a switch in a private network.

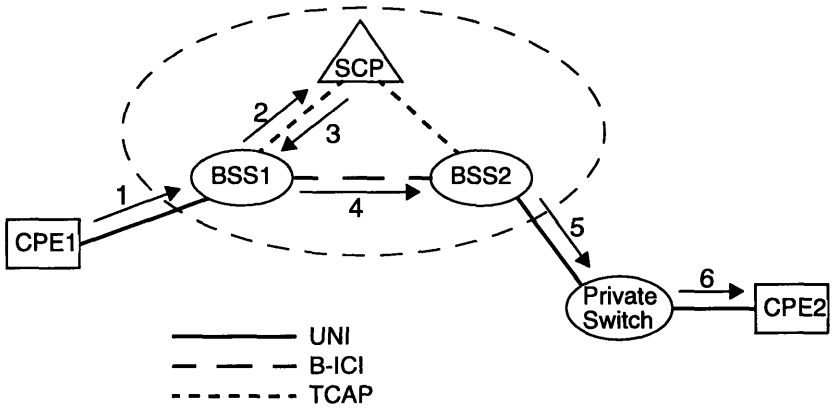


Figure 8-1. Call Flow for AESA-to-E.164 Translation — IN-Capable End Office

1. CPE1 sends a SETUP message to BSS1 with the AESA of CPE2. The AESA can be sent either in the *Called Party Number IE* (Information Element) or in the *Called Party Subaddress IE* of the SETUP message. If the *Called Party Subaddress* is used, the *Called Party Number* would be left empty.

2. BSS1 may receive the setup in one of two ways, as described in step 1 above.

A. The called AESA is sent to BSS1 in the *Called Party Number IE*. BSS1 analyzes the address and recognizes that the *Called Party Number* in the SETUP message is not an E.164 number, it is an AESA.

B. The called AESA is sent to BSS1 in the *Called Party Subaddress IE* and the *Called Party Number IE* is left blank. BSS1 knows to look for an AESA in the *Called Party Subaddress* because of the empty *Called Party Number.*

BSS1 may check to see if the address is an E.164 AESA. If it is an E.164 AESA, processing may skip to step 4 below. However, for certain implementations BSS1 may send out a query on all received AESAs. BSS1 then sends a query to the SCP with the called AESA.

3. The SCP responds with the public network E.164 address of the destination private network's E.164 address. It would likely also send the AESA back to BSS1 to be included in the IAM of step 4.

4. Routing on the E.164 address of the private network, BSS1 sends an IAM to BSS2 with the E.164 address of the private network as the *Called Party Number.* The AESA of CPE2 is included either in the *Called Party Subaddress* or in the *AESA for Called Party* parameter of the IAM. See section 2.8.2 for a description of the *AESA for Called Party* parameter.

5. BSS2 puts the AESA from either the *Called Party Subaddress* or the *AESA for Called Party* parameter back into the *Called Party Number IE* and sends a SETUP message to the private switch. The E.164 address of the private network is no longer needed.[1]

1. This second translation (moving the AESA from the *Called Party Subaddress* or *AESA for Called Party* parameter back into the *Called Party Number IE*) can be done through another query to the SCP. However, the only thing that must be done is fill the *Called Party Number IE* with the AESA that was carried in the IAM of step 4. This is a simple operation that would be done more efficiently by BSS2 than through a query to the SCP.

6. The private switch sends a SETUP message to CPE2 with the AESA in the *Called Party Number IE*. If there is more than one switch in the private network, the call is routed using P-NNI procedures and CPE2's AESA.

As described above, the originating SETUP message may contain the called AESA in either the *Called Party Number* or the *Called Party Subaddress*. Since the AESA is in fact the called party number, carrying it in the *Called Party Number IE* would make sense, and is the method assumed by the B-ICI specification. However, as an initial implementation it may be easier for BSS1 to recognize that it has received an AESA by looking for an empty *Called Party Number* rather than having to analyze the *Called Party Number* to determine the type of address it contains. Either the CPE would have the necessary intelligence to send the called AESA in the *Called Party Subaddress IE* or the user would make a manual selection to do so.

8.2 Details of Address Translation Service

The UNI, as well as access signalling necessary for an ATM call setup are defined in the ATM Forum's *User-Network Interface Specification* [6]. Bellcore's GR-1111-CORE, *Broadband Access Signaling Requirements* [2] defines the access signaling for the LECs. While both documents are very well aligned, GR-1111-CORE contains some specifics for public UNIs. Public network-network signalling is defined by the Broadband Inter Carrier Interface (B-ICI) specification [7].

This section discusses messages and parameters needed to carry out the address translations described in section 8.1. IN messages and parameters are based on those of narrowband IN, described in section 3. Necessary changes for IN to support broadband are stated.

8.2.1 The Originating SETUP

First, the calling user sends a SETUP message to the public network containing the address of the called party. If the called party address is:

- E.164: The originating switch routes the call normally.
- AESA E.164: The called party address contains enough information for the originating BSS to route the call. The BSS can route the call on the E.164 address and does not have to query any SCP to obtain routing information. The steps described in section 8.2.2 for querying an SCP can be skipped. However, certain implementations of ATM address translation may require all AESAs to be sent to an SCP for translation. This would relieve the BSS from having to differentiate between the tree types of AESAs and of having to extract the E.164 portion of an AESA E.164 address.
- ICD or DCC AESA: The originating BSS does not have enough information to determine where to route the call. It therefore queries an SCP to obtain the E.164 address that should be used to send the call to the private network that serves the called party.

GR-1111-CORE contains conditional requirements in support of allowing AESAs in the *Called Party Number IE* of a SETUP message, in the same way as AESAs are allowed in UNI 3.1.[2] This means that if public networks wish to support AESAs (at least in the initial call setup), and if their equipment can support AESAs, then they will do so. These conditional requirements allow AESAs to be supported as described in UNI 3.1. The BSS would therefore have to recognize that the *Called Party Number* is an AESA and it would query an SCP to obtain routing information.

2. These conditional requirements are in section 8.5.11.2 of GR-1111-CORE.

The calling user can otherwise send the called party AESA in the *Called Party Subaddress IE* of the SETUP message, leaving the *Called Party Number IE* empty or simply not sending it in the SETUP message. UNI 3.1 states that the *Called Party Subaddress IE* is used to convey an ATM address in AESA format across a public network which supports only E.164 addresses.[3] When the BSS receives a SETUP message with an empty or missing *Called Party Number IE* it would know that the *Called Party Subaddress* should contain the called AESA. The BSS then queries an SCP to obtain routing information. The *Called Party Number IE* is mandatory, as defined by UNI 3.1.[4] Therefore, although detecting the non-existence of a *Called Party Number IE* may be a simple task for the BSS, this method conflicts with current UNI standards.

GR-1111-CORE supports the structure of the *Called Party Subaddress IE* as described in UNI 3.1, which supports AESAs. However, it also says that the *Called Party Subaddress IE* is not examined by the BSS.[5] This is not stated as a requirement, but it is in the text. As long as the BSS can recognize that the called address is an AESA simply by the lack of a *Called Party Number* along with the existence of a *Called Party Subaddress*, it does not have to analyze the AESA. If the BSS is to analyze the AESA to determine whether it is an E.164 AESA, this part of GR-1111 would need to be changed.

Note that depending on the implementation the BSS may not query an SCP when it receives an E.164 AESA in a SETUP message.

3. Section 5.4.5.12 of the UNI 3.1 specification.
4. Section 5.3.1.7 of the UNI 3.1 specification.
5. Section 8.5.12 of GR-1111-CORE.

8.2.2 Querying the SCP

The originating BSS now has the called party address, whether it was sent in the *Called Party Number IE* or in the *Called Party Subaddress IE* of the SETUP message. If the called party address is:

- E.164: No query is needed. The BSS routes the call normally, skips the steps described in this and the following section, and continues with the procedures described in section 8.2.4.
- AESA E.164: If the implementation is such that all AESAs are sent to an SCP for translation, then a query is sent to the SCP. Otherwise, the procedures described in this and the following section are skipped and the call continues as described in section 8.2.4.
- ICD or DCC AESA: A query containing the called party address is sent to an SCP in order to obtain an E.164 address that the originating BSS should use to route the call to the private network that serves the called party.

The BSS, detecting the that the called party address is an AESA, halts call processing and sends a query to the SCP containing the AESA.

A trigger in the *ANALYZE_INFORMATION* PIC (see section 3.3) can be used to detect that the called party number is an AESA, thus causing the originating BSS to send an *Info_Analyzed* TCAP message (see section 3.4) to an SCP. BSSs would have to be able to trigger on AESAs.

If the called party AESA is sent to the network in the *Called Party Subaddress IE*, as described in section 8.2.1, then the BSS would trigger on the existence of a *Called Party Subaddress IE* along with no *Called Party Number IE* and then send an *Info_Analyzed*

TCAP message to the SCP. It may be easier, at least in early implementations, to trigger on the empty *Called Party Number IE* and on the existence of a *Called Party Subaddress IE* rather than on a field of an information element (i.e., the type of address).

The choice of triggering in the *ANALYZE_INFORMATION* PIC and sending an *Info_Analyzed* TCAP message, as described in the above paragraph, assumes that a user of the public network may send a SETUP message with either an E.164 address or an AESA such that the originating BSS has to distinguish between them in order to decide whether or not to query an SCP for a translation. If in advance a translation from AESA to E.164 is known to be needed then the BSS can instead trigger in the *COLLECT_INFORMATION* PIC and send an *Info_Collected* TCAP message. Thus, the BSS does not have to analyze the called party address information it receives in the SETUP message just to find that it must query an SCP anyway. The need for a translation can be known in advance if, for example, several of an organization's private networks using ICD or DCC AESAs are known to only set up calls between each other.

SETUP messages from a private network can all be assumed to contain AESAs. It may, however, still be necessary for the originating BSS in the public network to distinguish between E.164 AESAs and ICD or DCC AESAs in order to determine whether an SCP query is needed.

Local Number Portability will likely affect the decision of which TCAP message to use, *Info_Collected* vs. *Info_Analyzed.* If many or all numbers become portable in the future, switches may have to query an SCP regardless of the addressing scheme of the called party address, in which case *Info_Collected* would be used. However, if different SCPs are queried depending on various criteria, then *Info_Analyzed* would more likely be used because the BSS would have to analyze the called party address sufficiently to decide which SCP to query.

When the originating BSS sends its query to the SCP it includes the called AESA. If the query is sent in an *Info_Analyzed* TCAP message the *CalledPartyID* [6] parameter can be used to pass the called AESA to the SCP.

If instead the query is sent in an *Info_Collected* TCAP message, the called party address can be sent to the SCP in the *CollectedAddressInfo* parameter, without prior interpretation by the BSS.

8.2.3 Response from the SCP

The SCP translates the called party AESA into the E.164 address of the private network to which the call should be sent in order to reach the called party. It then sends a response to the BSS's query. The response contains the E.164 address along with the original called AESA. The called AESA is sent back to the BSS in order for the BSS to send it through the public network to the destination private network. The SCP's response is sent in an *Analyze_Route* TCAP message. The *Analyze_Route* message indicates to the BSS that it must route the call to the destination indicated in the message.

The SCP sends the E.164 address of the destination private network to the BSS in the *CalledPartyID* parameter of the *Analyze_Route* message. The called party AESA is also be sent back to the BSS in the *Analyze_Route* message. This message needs a parameter to carry the called party AESA to the BSS in this TCAP message — a parameter such as this one does not exist in narrowband IN. This parameter can be based on the new B-ISUP *AESA for Called Party* parameter.[7] If the parameter is based on or is identical to the B-ISUP *AESA for Called Party* parameter then the called AESA from the *Analyze_Route* message

6. See section 3.4 for a brief description of TCAP message parameters

7. The *AESA for Called Party* parameter is defined in ITU-T draft Recommendation Q.2726.1 and described briefly in section 2.8.2 of this thesis.

can be copied directly into the *AESA for Calling Party* parameter in the IAM sent through the network.

The parameter could otherwise be based on the B-ISUP *Called Party Subaddress*. However, since ITU standards seem to be moving towards support of the *AESA for Called Party* parameter, adopting this parameter seems better than adopting the *Called Party Subaddress* parameter.

8.2.4 Network Setup: The IAM

The originating BSS now has the E.164 address where the call is to be sent in the public network and it also has the called AESA, which it may not understand but nevertheless should send through the public network to the destination private network. The E.164 address was obtained either from an E.164 AESA in the *Called Party Number* of the SETUP message or from the *CalledPartyID* of the *Analyze_Route* message sent from the SCP. The called party AESA was sent back from the SCP in the new parameter described above (like the *AESA for Called Party* parameter) of the *Analyze_Route* from the SCP.

The BSS is now ready to send an *Initial Address Message* (IAM) through the network in order to send the call to the destination. It places the E.164 address in the *Called Party Number* parameter of the IAM, and places the called AESA in either the *Called Party Subaddress* or the *AESA for Called Party* parameter of the IAM.

The *Called Party Number* parameter supports E.164 addresses — this is its normal use. The *Called Party Subaddress* can support AESAs, as can the *AESA for Called Party* parameter. Although both parameters can support AESAs, in the long term it may be desirable to use the *AESA for Called Party* parameter since it was recently defined specifically for the

purpose of carrying AESAs through the public network. In addition, using the *AESA for Called Party* parameter leaves the *Called Party Subaddress* free for other uses.

If the BSS finds that the call should terminate at one of its own UNIs then an IAM is not needed and processing continues as described in section 8.2.5.

8.2.5 The Terminating SETUP

The terminating BSS now has an E.164 address to which it should terminate the call and the AESA which it should send across the UNI. It places the called AESA from either the *Called Party Subaddress* or the *AESA for Called Party* of the IAM back into the *Called Party Number IE* of the SETUP message and sends this SETUP message across the UNI to the private network. The E.164 number is no longer needed. If the E.164 number was extracted from an E.164 AESA, the E.164 portion will still be in the AESA. If the AESA is non-E.164 then the private network will most likely have no use for the E.164 address.

The public BSS could instead send the E.164 address in the *Called Party Number* of the SETUP message and the AESA in the *Called Party Subaddress*. The private network would then most likely discard the E.164 address and place the AESA back in the *Called Party Number* before routing the call in its network. However, if the public network is going to take care of the AESA-to-E.164 translation it would make sense if the call were presented to the private network as a call to the AESA, just as it was presented to the public network in the first place, i.e. the AESA in the *Called Party Number IE* of the SETUP message.

8.3 IN Through Access Tandem

An alternative IN architecture is one where only a limited number of switches in the network are IN-capable. On initial deployment of Broadband IN there may be only one BSS in the network with IN functionality. Calls that require IN operations would be forwarded to an IN-capable BSS which would act as an Access Tandem.

Given the sophistication of ATM switches, it does not seem very difficult to add IN functionality to a public switch that will likely have B-ICI built into it anyway. But this scenario is briefly presented here to show how it might work.

A typical call through the public network using an Access Tandem for an AESA-to-E.164 translation is shown in Figure 8-2 and described below. As in section 8.1, CPE1 may be either an end user or a private network and CPE2 is connected to a switch in a private network. BSS1, BSS2, and BSS3 are in the public network. BSS2 is an Access Tandem.

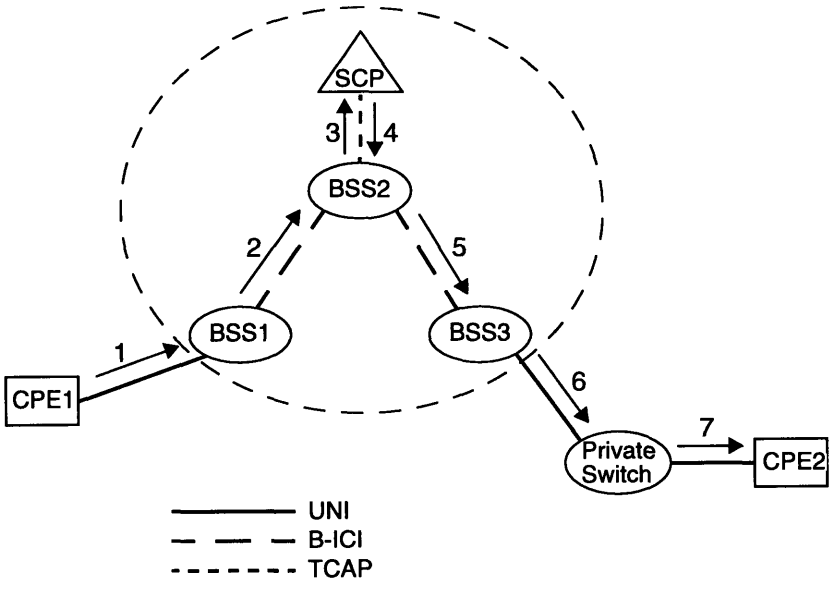


Figure 8-2. Call Flow for AESA-to-E.164 Translation — Access Tandem

1. CPE1 sends a SETUP message to BSS1 with the AESA of CPE2 either in the *Called Party Number* or *Called Party Subaddress IE*, as in the call flow of section 8.1.
2. BSS1 recognizes that the called party address is an AESA that requires translation and sends the call to BSS2. As in section 8.1 BSS1 may look for either an AESA in the *Called Party Number* or an empty *Called Party Number* of the SETUP message, which would indicate the presence of an AESA in the *Called Party Subaddress*. BSS1 then sends the AESA of CPE2 to its Access Tandem, BSS2, in either the *Called Party Subaddress* or the *AESA for Called Party* parameter of an IAM. Note that the *Called Party Number* of an IAM cannot carry an AESA. In either case, the *Called Party Number* would be left blank, as the E.164 address will not be known until the SCP is queried. If the AESA is in E.164 format it could be possible for BSS1 to extract the E.164 information from an E.164 AESA, place the appropriate parameters in an IAM and send the call directly to BSS3 as in step 4 of section 8.1.
3. BSS2 recognizes that the called party number is an AESA that requires translation and sends a query to the SCP with the AESA of CPE2.
4. The SCP responds with the E.164 address of the destination private network and also sends back the called AESA to be carried through to the private network.
5. BSS2 routes the call using the E.164 address obtained from the SCP and sends an IAM to BSS3 with the E.164 address in the *Called Party Number* parameter and the AESA in either the *Called Party Subaddress* or the *AESA for Called Party* parameter.
6. BSS3 sends a SETUP message to the destination private network with the AESA in the *Called Party Number IE*. The E.164 address is no longer needed.
7. The private network routes the call using the AESA and sends a SETUP message to CPE2.

Note that in this architecture all calls that require address translation would have to go through BSS2, the Access Tandem. This may not be a problem for an initial small deployment, but scalability will be a problem if all calls are actually routed through the Access Tandem.

Call processing in this scenario is very similar to that of section 8.2 where end offices are IN-capable. The main difference is that in this scenario the originating BSS sends the call to an Access Tandem to take care of the ATM address translation.

Once the originating BSS receives a SETUP message from a user and recognizes that the *Called Party Number* is not an E.164 number and may be an AESA, it sends the call to its Access Tandem as a default routing. The IAM should contain the called AESA in either the *Called Party Subaddress* or the *AESA for Called Party* parameter, as described in section 8.2.4. The originating BSS would not have any information with which to populate the *Called Party Number* parameter of the IAM it sends to the Access Tandem. However, the ATM Forum's B-ICI Specification[8] and GR-1417-CORE[9] both specify that the *Called Party Number* parameter shall be included in the IAM. Therefore, not including the *Called Party Number* parameter may be a problem for existing BSSs. An indication would need to be included in the IAM to indicate to the Access Tandem that the IAM contains only an untranslated AESA.

8.4 Using Narrowband AIN for ATM Address Translations

In order to set up an ATM address translation service as quickly as possible, before Broadband IN is readily available, Narrowband AIN could be used (see section 3 for a brief

8. Requirement 7.35 of the ATM Forum's B-ICI specification.

9. Requirement R-96 of Bellcore's GR-1417-CORE.[5]

description of narrowband AIN). Most of the changes required are to support parameters specific to ATM.

8.4.1 Querying the SCP

Two messages can be used to query the SCP. These are *Info_Collected* and *Info_Analyzed*. *Info_Collected* would be used if the SCP is to be queried for all set up requests. This could be the case for a private network which always requests connections to AESAs. *Info_Analyzed* would be used if the SCP is to be queried only for some set up requests. The BSS would determine whether or not to query the SCP and then, if a query is needed, an *Info_Analyzed* message would be sent. Note that the *Info_Collected* message is sent as soon as the information from the user is collected by the BSS while the *Info_Analyzed* message is sent after the BSS analyzes the address information provided by the user.

Several of the parameters that could be used to convey AESAs to an SCP are defined following the format of the *AINDigits* parameter (see section 3.4). The format of *AINDigits* fits two "dialed digits" in each octet of digit information within the *AINDigits* parameter.[10] In order to allow *AINDigits* to carry AESAs the *AINDigits* format should allow each octet to carry an octet of the AESA. In addition, a code would need to be added for use in this parameter to identify the AESA addressing schemes.[11]

In the *Info_Collected* TCAP message the called AESA could be sent in the *CollectedAddressInfo* parameter. The limit of 15 digits held by this parameter, however,

10. Requirement R-205 of Bellcore's GR-1299-CORE.

11. Requirement R-210 of Bellcore's GR-1299-CORE.

would need to be relaxed.[12] The *CollectedAddressInfo* parameter is defined to use the *AINDigits* parameter format with a range limit of 0-15 digits (0-8 octets of information).

Otherwise, the *CollectedDigits* parameter could be used. The *CollectedDigits* parameter is also defined to use the *AINDigits* parameter format.[13] In fact, it is defined with a range of 0-42 digits (up to 21 octets). Therefore without changing the format of the *CollectedDigits* parameter it can accommodate all 20 octets of an AESA. Each octet of digit information in the *CollectedDigits* parameter would be filled with an octet of the AESA.

Another option is to send the AESA in a *GenericAddress* field of a *GenericAddressList* parameter. However, the length of the *GenericAddress* is limited to be at most 9 octets.[14] This limitation would need to be relaxed for the *GenericAddress* to be used.

If instead of sending an *Info_Collected* message the BSS were to send an *Info_Analyzed* message, it could send the AESA in a *CalledPartyID* parameter. To support AESAs, however, the *CalledPartyID* would need to be expanded from it current limit of 15 digits.[15] There are also other limitations imposed on the *CalledPartyID* parameter that would have to be relaxed before an AESA could be carried by it.[16]

The *CollectedAddressInfo* parameter and *GenericAddress* of a *GenericAddressList* parameter could be used if the modifications described above, for the case of the *Info_Collected* message, are met.

12. Requirement R-311 of Bellcore's GR-1299-CORE.

13. Requirement R-315 of Bellcore's GR-1299-CORE.

14. Requirement R-343 of Bellcore's GR-1299-CORE.

15. Requirement R-295 of Bellcore's GR-1299-CORE.

16. Section 4.5.3.3 of Bellcore's GR-1299-CORE.

An *Info_Analyzed* message can also contain a *CollectedDigits* parameter. As explained above this parameter could carry a complete AESA.

It seems that the *CollectedDigits* parameter can be used to carry a called AESA to an SCP in either an *Info_Collected* or an *Info_Analyzed* TCAP message, whichever is chosen to be used. GR-1299-CORE defines how the *CollectedDigits* parameter should be used in a Narrowband AIN *Info_Collected*[17] TCAP message and in an *Info_Analyzed*[18] TCAP message.

8.4.2 Response from the SCP

The SCP would respond to the BSS using an *Analyze_Route* TCAP message. This message would contain the E.164 address that the BSS should use to route the call in the public network and the called AESA to be sent through the public network to the destination private network.

The E.164 address would be sent in the *CalledPartyID* parameter of the *Analyze_Route* message. As defined in GR-1299-CORE the *CalledPartyID* parameter supports E.164 addresses. No changes are needed.

No specific parameters of an *Analyze_Route* message can readily carry an AESA back to the switch. Therefore the *ExtensionParameter* may be used.[19] This parameter allows networks to define new parameters that are not defined in Intelligent Network standards. A new parameter may be defined for use in the *ExtensionParameter* for the SCP to send the called AESA back to the BSS. The BSS would then send the AESA transparently through

17. Requirement R-192 of Bellcore's GR-1299-CORE.

18. Requirement R-230 of Bellcore's GR-1299-CORE.

19. Conditional requirement CR-334 of Bellcore's GR-1299-CORE.

the public network. The new parameter could be based on either the B-ISUP *AESA for Called Party* or *Called Party Subaddress* parameters so as to simplify the BSSs processing of the *Analyze_Route* message. The BSS would simply copy the information into either an IAM *AESA for Called Party* of *Called Party Subaddress* parameter, whichever is used.

In lieu of the *ExtensionParameter* a *GenericAddress* of a *GenericAddressList* could be used, were it not for its length limit.

Alternatively, only the E.164 address may be sent back from the SCP (the AESA would not be sent back) in which case the BSS would need sufficient intelligence to store the AESA until it gets a response from the SCP and then send it through the network in an IAM.

8.5 Database Partitioning

The issue of how to partition ATM address translation databases will likely become an important one as both public and private ATM networks are deployed. Initially, the simplest method may be to set up one or several replicated databases. This centralized database structure, similar to that of 800 service, may be easy to administer initially. It may, however, impose scalability constraints. Technology is currently available to support very large databases that can respond to queries very quickly. For example, 800 number databases contain about 8 million entries and process about 900 calls per second. But the size to which broadband networks will grow is currently difficult to determine. Centralized databases can be further replicated so as to lighten their query processing load so long as the databases are correctly synchronized.

Another possible structure is a distributed structure like the one used for LIDB (Line Information Database) and for wireless HLRs and VLRs. This structure is much more

scalable than the centralized one. The issue of how the database should be partitioned and distributed would have to be addressed.

There could be one database per type of AESA. A BSS that requires a translation queries an SCP that specifically serves translation requests for the given type of AESA. There could be a database that serves translations from ICD AESAs and another that serves translations from DCC AESAs. Depending on how E.164 AESAs are assigned another database may be needed for this AESA format. If the E.164 addresses used in E.164 AESAs follow NANP guidelines then no translation is needed. However, if Ugly E.164 addresses or E.164 addresses whose assignment differs from assignments in the public network are used then a database may be needed for E.164 AESAs as well.

Databases for ICD and DCC AESAs could be further subdivided according to country, organization, or groups of countries or organizations. Since AESAs have a hierarchical address structure the databases could, in principle, be partitioned based on virtually any part of the address.

It would be desirable for the method of partitioning the ATM address translation database to allow fast query response times and ease of administration. Partitioning the database based on administrative domains may be possible. This would raise issues concerning sharing of data across administrative boundaries.

Although an in-depth study of how to partition the databases has not been done, partitioning based on a portion of the address may offer the fastest access to the desired database entry, thus offering the fastest query response time. SCPs containing these databases would need to be deployed in mated pairs to increase reliability

8.6 Logical Name Translations

The translation functions of Broadband Intelligent Networks could also be used for logical name translations, such as those carried out by the Internet Domain Name System (DNS). This would allow users to request connections by giving a name for the endpoint rather than its network address.

These logical name translations could be done much in the same way as the ATM address translations described above. However, new messages would have to be defined because current address information elements in existing messages cannot carry long strings of text. A call set up could be requested by a user specifying a called name rather than a called number. This name would be translated into an E.164 address, possibly accompanied by an AESA, and the call would progress as normal.

Actually implementing DNS in an Intelligent Network, however, may be a bit more tricky. DNS interactions are between a host and a name server. In Intelligent Networks, the lookup interactions are normally between a switch and a database, rather than between the user (or host) and the database. Implementing DNS per se would require having an address (the address of a name server) to which host resolvers would send queries. If IN SCPs are to be used for storing DNS information, a task which they could easily carry out, the name server would need a way to communicate with the SCPs. Name servers are simply programs, so by adding necessary logic to an SCP it could be turned into a DNS name server. Users, or their hosts, could communicate with name servers in SCPs through IN Non-Call Associated Messages (NCAS). These messages would be sent between the host and its switch, and the switch would then forward an appropriate message to and from the SCP. This issue requires further study.

9. Conclusions

This thesis studied the issue of setting up calls across the public ATM network when one terminal is in a private network, i.e. the called address is not a public E.164 address. Existing standards support calls set up into the public network using both public E.164 and AESA E.164 called addresses, however, there is currently no mechanism for a call to be set up in the public network to the other private address types (ICD or DCC AESA).

To support private addresses in call set up across a public network, a translation from the private address to the public address of the network where the user is located is needed. This public address is then used to route the call to the destination. Having considered pros and cons of translating inside and outside of the public network in section 6, translation inside the public network was seen as reasonable and advantageous. This choice was made because translating in the public network can offer all users of public and private networks the same service, and updating of the translation database can be done centrally by the public network as it is done in telephony. Approaches where the translation is done in private networks, jointly by public and private networks, or even by a separate entity somewhere in the network could also be taken. Little information was found on how these other approaches might work, so further investigation into them would be valuable.

Section 8 presented a method for offering ATM address translation service by Intelligent Networks. The basic building blocks of Intelligent Networks can be used. However, changes are needed in existing standards for Intelligent Networks to support B-ISDN. The main changes are that address parameters in IN TCAP messages have to be modified to support ATM addresses. In addition, subaddress-type parameters have to be added to support ATM subaddresses and to allow public networks to carry addresses other than the public network endpoint address through to the destination of the call.

Section 8 also discussed how Narrowband AIN can be used to offer an ATM address translation service. This may be desirable before standards for broadband IN are established. Existing narrowband AIN can be used to support the basic translation service. Necessary service logic would need to be added to Narrowband SCPs in order to support the service. However, since narrowband AIN was not designed to support broadband network services, the method described in this thesis is more of a "quick fix" because of the lack of broadband resources in narrowband AIN. A brief discussion of how ATM address translation databases might be set up is also included in section 8.

With the IN-based ATM address translation service in place, like the one presented in this thesis, users anywhere on an ATM network can set up calls to other users. The ability to set up calls anywhere in the ATM internetwork not only allows users to communicate between each other but it allows higher layer protocols to operate over ATM across multiple networks.

Discussion in this thesis focused on point-to-point call setup. This work needs to be extended to cover point-to-multipoint calls as well. Also, while our focus was primarily on carrying and translating the *called party number* in order to set up the call initially, other parameters such as the *calling party number* may also require extended network services and should be studied.

Glossary

AAL	ATM Adaptation Layer
AESA	ATM End System Address (private ATM address)
AIN	Advanced Intelligent Network
ANSI	American National Standards Institute
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
ATMARP	ATM Address Resolution Protocol
B-ICI	Broadband Inter Carrier Interface
B-ISDN	Broadband ISDN
B-ISUP	Broadband Integrated Services Digital Network User Part
BSS	Broadband Switching System
CCS	Common Channel Signaling
CPE	Customer Premises Equipment
DCC	Data Country Code
DNS	Domain Name System
HLR	Home Location Register
IAM	Initial Address Message
ICD	International Code Designator
IEC	Inter Exchange Carrier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IN	Intelligent Network
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITU-T	International Telecommunications Union -Telecommunication Standardization Sector.
LAN	Local Area Network
LANE	LAN Emulation
LE_ARP	LAN Emulation Address Resolution Protocol
LEC	Local Exchange Carrier

LIDB	Line Information Database
LIS	Logical IP Subnetwork
LLC	Logical Link Control
MAC	Medium Access Control
NANP	North American Numbering Plan
NBMA	Non-Broadcast Multiple Access
NHRP	Next Hop Resolution Protocol
NHS	Next Hop Server
NSAP	Network Service Access Point
OSI	Open Systems Interconnect
PIC	Point In Call
P-NNI	Private Network-to-Network Interface
POTS	Plain Old Telephony Service
PVC	Permanent Virtual Connection
RFC	Internet Request for Comments
SAAL	Signaling ATM Adaptation Layer
SCP	Service Control Point
SSP	Service Switching Point
SS7	Signaling System Number 7
STP	Signaling Transfer Point
SVC	Switched Virtual Connection
TCAP	Transaction Capabilities Application Part
TCP/IP	Transmission Control Protocol / Internet Protocol
UNI	User-Network Interface
VCI	Virtual Channel Identifier
VLR	Visitor Location Register
VPI	Virtual Path Identifier

References

1. Bellcore. GR-246-CORE, *Bellcore Specification of Signalling System Number 7 (SS7)*, Issue 1. December 1994.
2. Bellcore. GR-1111-CORE, *Broadband Access Signaling Generic Requirements*, Issue 1, Revision 1. October 1995.
3. Bellcore. GR-1298-CORE, *Advanced Intelligent Network (AIN) Switching Systems Generic Requirements, Issue 2*. December 1995.
4. Bellcore. GR-1299-CORE, *Advanced Intelligent Network (AIN) Switch - Service Control Point (SCP)/Adjunct Interface Generic Requirements*, Issue 2, Revision 2. June 1995.
5. Bellcore. GR-1417-CORE, *Broadband Switching System SS7 Requirements Using BISUP*, Issue 1, Revision 1. October 1994.
6. ATM Forum. *ATM User-Network Interface (UNI) Specification,* Version 3.1. September 1994.
7. ATM Forum. *ATM Forum 94-180R5: A Draft of the B-ICI Specification Document,* Version 2.0. February 1995.
8. ATM Forum. *ATM Forum 94-0471R7: P-NNI Draft Specification.* March 1995.
9. ATM Forum. *ATM Forum 94-0035R9: LAN Emulation Over ATM Specification,* Version 1.0. January 1995.
10. ITU-T Recommendation E.163, *Numbering Plan for the International Telephone Service.* 1989.
11. ITU-T Recommendation E.164, *Numbering Plan for the ISDN Era.* 1989.
12. ITU-T Recommendation E.166/X.122, *Numbering Plan Interworking for the E.164 and X.121 Numbering Plans.* 1992.
13. ITU-T Recommendation I.321, *B-ISDN Protocol Reference Model and its Application.* 1991.
14. ITU-T Recommendation I.361, *B-ISDN ATM Layer Specification.* June 1992.
15. ITU-T Recommendation I.362, *B-ISDN ATM Adaptation Layer (AAL) functional description.* March 1993.
16. ITU-T Recommendation I.363, *B-ISDN ATM adaptation layer (AAL) specification.* March 1993.

17. ITU-T Recommendation X.31, *Support of Packet Mode Terminal Equipment by an ISDN*. November 1988.

18. ITU-T Recommendation X.75, *Packet-switched signalling system between public networks providing data transmission services*. March 1993.

19. ITU-T Recommendation X.121, *International Numbering Plan for Public Data Networks*. September 1992.

20. ITU-T Recommendation X.213, *Information technology - Network service definition for open systems interconnection*. September 1992.

21. ITU-T Recommendation X.500, *Information technology - Open Systems Interconnection - The directory: overview of concepts, models, and services*. November 1993.

22. ITU-T draft Recommendation Q.2726.1, *B-ISUP, ATM End System Address (AESA) for Calling and Called Party*. October 1995.

23. ITU-T Recommendation Q.2761, *BISDN, BISDN User Part - Functional Description*. March 1994.

24. ITU-T Recommendation Q.2762, *BISDN, BISDN User Part - General Functions of Messages and Signals*. March 1994.

25. ITU-T Recommendation Q.2763, *BISDN, BISDN User Part - Formats and Codes*. June 1994.

26. ITU-T Recommendation Q.2764, *B-ISDN, B-ISDN User Part - Basic Call Procedures*. June 1994.

27. ITU-T draft Recommendation Q.2931, *B-ISDN, Digital Subscriber Signalling System No. 2 (DSS 2), User-Network Interface (UNI), Layer 3 Specification for Basic Call/Connection Control*. May 1995.

28. RFC 826, *An Ethernet Address Resolution Protocol -- or -- Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*, David C. Plummer. November 1982.

29. RFC 903, *A Reverse Address Resolution Protocol*, Finlayson et al. June 1984.

30. RFC 1034, *Domain Names — Concepts and Facilities*, P. Mockapetris. November 1987.

31. RFC 1035, *Domain Names — Implementation and Specification*. November 1987.

32. Mockapetris, P. RFC 1101, *DNS Encoding of Network Names and Other Types.* April 1989.

33. Everhart, C., et al. RFC 1183, *New DNS RR Definitions.* October 1990.

34. Heinanen, Juha. RFC 1483, *Multiprotocol Encapsulation Over ATM Adaptation Layer 5.* July 1993.

35. Laubach, M. RFC 1577, *Classical IP and ARP over ATM.* January 1994.

36. Manning, B., et al. RFC 1706, *DNS NSAP Resource Records.* October 1994.

37. Katz, David, and Piscitello, David. *NBMA Next Hop Resolution Protocol (NHRP),* Internet Draft. May 1995.

38. ISO 3166, *Codes for the representation of names of countries.* 1993.

39. ISO 6523, *Data interchange -- Structures for the identification of organizations.* 1984.

40. ISO 8348, *Information technology -- Open Systems Interconnection -- Network Service Definition.* 1993.

41. ISO 10589, *Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service.* 1992.

42. Armitage, Grenville John. *The Application of Asynchronous Transfer Mode to Multimedia and Local Area Networks.* Department of Electrical and Electronic Engineering, University of Melbourne, Australia. January 1994.

43. Klessig, Bob, 3Com Corporation ATM Forum Ambassador. *Intermediate ATM,* 53 Bytes. The ATM Forum Newsletter, Volume 2 Issue 3. September 1994. http://www.atmforum.com/atmforum/53bytes-backissues/53bytes-0994-4.html (URL last visited 5/6/96).

44. Alles, Anthony, ATM Product Line Manager, Cisco Systems, Inc. *ATM Internetworking.* May 1995. http://cio.cisco.com/warp/public/614/12.html (URL last visited 5/7/96).

45. Ibe, Oliver C. *A Framework for an Intelligent Network Call Model for ATM Networks.* Laboratory for Information and Decision Systems, MIT. 1993.

46. Radicati, Sara. *X.500 Directory Services: Technology and Deployment.* Van Nostrand Reinhold. 1994.

47. Acampora, Anthony S. *An Introduction to Broadband Networks: LANs, MANs, ATM, B-ISDN, and Optical Networks for Integrated Multimedia Telecommunications*. Plenum Press. 1994.

48. Dickie, Mark. *Routing in Today's Internetworks*. Van Nostrand Reinhold. 1994.

49. Albitz, Paul and Liu, Cricket. *DNS and BIND in a Nutshell*. O'Reilly & Associates, Inc. 1992.