X/Open CAE Specification

Multiprotocol Transport Networking (XMPTN): Address Mapper

X/Open Company Ltd.

X/Open CAE Specification

Multiprotocol Transport Networking (XMPTN): Address Mapper

ISBN: 1-85912-101-2

X/Open Document Number: C520

Published by X/Open Company Ltd., U.K.

Any comments relating to the material contained in this document may be submitted to X/Open at:

X/Open Company Limited
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:

XoSpecs@xopen.org

Contents

List of Figures

List of Tables

Preface

X/Open

X/Open is an independent, worldwide, open systems organisation supported by most of the world's largest information systems suppliers, user organisations and software companies. Its mission is to bring to users greater value from computing, through the practical implementation of open systems.

X/Open's strategy for achieving this goal is to combine existing and emerging standards into a comprehensive, integrated, high-value and usable open system environment, called the Common Applications Environment (CAE). This environment covers the standards, above the hardware level, that are needed to support open systems. It provides for portability and interoperability of applications, and so protects investment in existing software while enabling additions and enhancements. It also allows users to move between systems with a minimum of retraining.

X/Open defines this CAE in a set of specifications which include an evolving portfolio of application programming interfaces (APIs) which significantly enhance portability of application programs at the source code level, along with definitions of and references to protocols and protocol profiles which significantly enhance the interoperability of applications and systems.

The X/Open CAE is implemented in real products and recognised by a distinctive trade mark — the X/Open brand — that is licensed by X/Open and may be used on products which have demonstrated their conformance.

X/Open Technical Publications

X/Open publishes a wide range of technical literature, the main part of which is focussed on specification development, but which also includes Guides, Snapshots, Technical Studies, Branding/Testing documents, industry surveys, and business titles.

There are two types of X/Open specification:

- *CAE Specifications*

  CAE (Common Applications Environment) specifications are the stable specifications that form the basis for X/Open-branded products. These specifications are intended to be used widely within the industry for product development and procurement purposes.

  Anyone developing products that implement an X/Open CAE specification can enjoy the benefits of a single, widely supported standard. In addition, they can demonstrate compliance with the majority of X/Open CAE specifications once these specifications are referenced in an X/Open component or profile definition and included in the X/Open branding programme.

  CAE specifications are published as soon as they are developed, not published to coincide with the launch of a particular X/Open brand. By making its specifications available in this way, X/Open makes it possible for conformant products to be developed as soon as is practicable, so enhancing the value of the X/Open brand as a procurement aid to users.

- *Preliminary Specifications*

  These specifications, which often address an emerging area of technology and consequently are not yet supported by multiple sources of stable conformant implementations, are released in a controlled manner for the purpose of validation through implementation of products. A Preliminary specification is not a draft specification. In fact, it is as stable as X/Open can make it, and on publication has gone through the same rigorous X/Open development and review procedures as a CAE specification.

  Preliminary specifications are analogous to the *trial-use* standards issued by formal standards organisations, and product development teams are encouraged to develop products on the basis of them. However, because of the nature of the technology that a Preliminary specification is addressing, it may be untried in multiple independent implementations, and may therefore change before being published as a CAE specification. There is always the intent to progress to a corresponding CAE specification, but the ability to do so depends on consensus among X/Open members. In all cases, any resulting CAE specification is made as upwards-compatible as possible. However, complete upwards-compatibility from the Preliminary to the CAE specification cannot be guaranteed.

In addition, X/Open publishes:

- *Guides*

  These provide information that X/Open believes is useful in the evaluation, procurement, development or management of open systems, particularly those that are X/Open-compliant. X/Open Guides are advisory, not normative, and should not be referenced for purposes of specifying or claiming X/Open conformance.

- *Technical Studies*

  X/Open Technical Studies present results of analyses performed by X/Open on subjects of interest in areas relevant to X/Open's Technical Programme. They are intended to communicate the findings to the outside world and, where appropriate, stimulate discussion and actions by other bodies and the industry in general.

- *Snapshots*

  These provide a mechanism for X/Open to disseminate information on its current direction and thinking, in advance of possible development of a Specification, Guide or Technical Study. The intention is to stimulate industry debate and prototyping, and solicit feedback. A Snapshot represents the interim results of an X/Open technical activity. Although at the time of its publication, there may be an intention to progress the activity towards publication of a Specification, Guide or Technical Study, X/Open is a consensus organisation, and makes no commitment regarding future development and further publication. Similarly, a Snapshot does not represent any commitment by X/Open members to develop any specific products.

Versions and Issues of Specifications

As with all *live* documents, CAE Specifications require revision, in this case as the subject technology develops and to align with emerging associated international standards. X/Open makes a distinction between revised specifications which are fully backward compatible and those which are not:

- a new *Version* indicates that this publication includes all the same (unchanged) definitive information from the previous publication of that title, but also includes extensions or additional information. As such, it *replaces* the previous publication.

- a new *Issue* does include changes to the definitive information contained in the previous publication of that title (and may also include extensions or additional information). As such, X/Open maintains *both* the previous and new issue as current publications.

Corrigenda

Most X/Open publications deal with technology at the leading edge of open systems development. Feedback from implementation experience gained from using these publications occasionally uncovers errors or inconsistencies. Significant errors or recommended solutions to reported problems are communicated by means of Corrigenda.

The reader of this document is advised to check periodically if any Corrigenda apply to this publication. This may be done in any one of the following ways:

- anonymous ftp to ftp.xopen.org
- ftpmail (see below)
- reference to the Corrigenda list in the latest X/Open Publications Price List.

To request Corrigenda information using ftpmail, send a message to ftpmail@xopen.org with the following four lines in the body of the message:

```
open
cd pub/Corrigenda
get index
quit
```

This will return the index of publications for which Corrigenda exist. Use the same email address to request a copy of the full corrigendum information following the email instructions.

This Document

This document is an X/Open CAE Specification. It defines the XMPTN Address Mapper, which provides dynamic address mapping services to nodes in a mixed transport protocol network. The XMPTN Address Mapper gives a general solution for address mapping which would otherwise have to be provided by multiple protocol-specific address mapping schemes.

X/Open MultiProtocol Transport Networking (XMPTN) architecture supports mixed transport protocol networking, enabling application programs that were designed to operate over one transport protocol - such as SNA, NetBIOS, OSI, or TCP/IP - to run over other transport networks. With XMPTN, existing applications can still run when the transport protocol for which they were written is exchanged for another one, and applications written for one transport can be introduced into networks which use other transports.

X/Open has published an XMPTN Architecture Guide which explains the role of address mapper in XMPTN. The associated XMPTN Access Node is defined in a separate X/Open specification, and the formats of message and data structures used by XMPTN are defined in the associated XMPTN Data Formats specification. X/Open offers a package of all four XMPTN documents, in Document Set T504.

This specification does not prescribe any specific implementation of MPTN.

Audience

This document is intended primarily for use by implementors of the MPTN Address Mapper functionality who wish to conform to the X/Open MPTN formats and protocols specification. It will also be of interest to diagnosticians who interpret these formats when analysing line flows, and to others who may wish to learn about the MPTN architecture from the data formats.

Network designers, network managers, or application program vendors who are interested in mixed protocol networking, as addressed by the MPTN architecture, are referred to the X/Open **Multiprotocol Transport Networking (MPTN) Architecture Guide**(see **Referenced Documents** on page x).

Structure

This specification is organised as follows:

- Part 1:
  - — Chapter 1 introduces MPTN and presents the terminology used in the XMPTN architecture.
  - — Chapter 2 describes the design and configuration requirements for XMPTN Address Mapping.
  - — Chapter 3 gives a brief overview of XMPTN Address Mapping Services.
- Part 2:
  - — Chapter 4 gives an overview of the Address Mapper protocol.
  - — Chapter 5 contains all the protocols that are used between the XMPTN Address Mapper and an XMPTN Access Node.
- Part 3:

  Appendix A provides supplementary information on protocol flow notation.

Trade Marks

UNIX® is a registered trade mark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X/Open® is a registered trade mark, and the ‘‘X’’ device is a trade mark, of X/Open Company Limited.

Referenced Documents

The following documents are referenced in this Specification:

IP Address

Interworking with TCP/IP: Principles, Protocols, and Architecture, Second Edition, D.E.Comer, published by Prentice Hall, Englewood Cliffs, NJ, USA, 1991.

XMPTN Access Node

X/Open CAE Specification, Multiprotocol Transport Networking (XMPTN): Access Node (ISBN: 1-85912-106-3, C521).

XMPTN Architecture Guide

X/Open Guide, December 1995, Multiprotocol Transport Networking (XMPTN) Architecture (ISBN: 1-85912-116-0, G506).

XMPTN Data Formats

X/Open CAE Specification, Multiprotocol Transport Networking (XMPTN): Data Formats (ISBN: 1-85912-111-X, C522).

X/Open CAE Specification

Part 1:
MPTN Address Mapper Overview

X/Open Company Ltd.

Chapter 1

Introduction

1.1 MPTN Architecture Terminology

This section presents the terminology used by MPTN and particular terms that are used in the Address Mapper.

In this document, *networking* means providing a relaying and routing service. When that service is provided at the transport and all lower layers, the resulting communication service is referred to as *transport networking* and an implementation of such transport networking is called a *transport network*.

The term *transport user* means application programs and application-support functions. The *transport user address* is the address in the transport user's native format. The term *transport provider* means a provider of communication service at the transport layer; to be more precise, it means providing the service at the transport/network layer, including subnetworking services. The *transport provider address* is the address as known by the transport provider (which would be non-native if the transport provider were of a different protocol family than the transport user). A transport provider uses one *transport protocol* to govern the exchange of information between nodes, thus providing a *transport network* of that type.

The boundary between the transport user and the transport provider is the *transport layer protocol boundary* (TLPB).

When discussing Address Mapper capabilities, references to transport users refer to functions that take place above the TLPB, while references to transport providers indicate activities that occur below the TLPB.

The terms *native* and *non-native* describe a vertical relationship between a transport user and its corresponding transport provider. With respect to a particular transport user, a transport provider that uses the address type and transport characteristics assumed in the transport-user design is native to that transport user. A transport provider that doesn't provide that address type and those transport characteristics is non-native. The Address Mapper function provides the address resolution capabilities that are required when a transport user requires the services of a particular transport provider that is non-native to that transport user.

A *single-protocol transport network* (*SPTN*) consists of a group of nodes that are physically connected and that implement the same transport protocol. These nodes may be connected by protocol-specific gateways, such as IP routers in a TCP/IP network or SNA network interconnection (SNI) nodes in an SNA network.

A *multiprotocol transport network* (*MPTN*) is a confederation of SPTNs, each of which has its own transport protocol.

The term *registration* concerns identification to the Address Mapper of a transport user address and one or more associated transport provider addresses. When the Address Mapper stores this relationship internally for future reference, the address is *registered*. The information stored by the Address Mapper; that is, the transport user address together with one or more associated transport providers, is called the *registered address pair*.

When an Access Node wants to find the location of a prospective partner, it uses the *address resolution* capabilities of MPTN to resolve the transport user address to a transport provider address.

A *transaction* in the Address Mapper protocol is a well-defined set of flows that performs a single activity, such as registering one set of address pairs. The transaction may span several ABM Commands. It is identified by a unique transaction ID.

The acronym *ABM* refers to the Address Mapper.

The acronym *AMS* refers to the Address Mapper Server. It is the component of MPTN that performs registration and resolution services on behalf of Address Mapper Clients.

The acronym *AMC* refers to the Address Mapper Client. It is the component of MPTN in the Access Node that contains transport users that need to be registered at the Address Mapper Server.

MPTN *multicast* allows a transport user to originate a datagram to be sent to more than one destination. We distinguish between inherent multicasting, wherein the transport provider has the capability to multicast data, and native multicasting, where the transport user and the transport provider protocols match and the protocol has inherent multicast capability.

1.2 MPTN Address Mapping Services

MPTN defines a service called the Address Mapper to provide mapping services to users. The Address Mapper Server is a transport user that communicates with Access Nodes to provide an address registration and management service. ABM commands are defined that allow an Access Node to register an address mapping by saving the association between the transport user's address and its corresponding transport provider's address(es), as well as other attributes associated with the transport user and/or provider(s). This capability enables MPTN to decouple a transport user address from the transport provider address.

The transformations provided by MPTN allow its user to specify addresses in its own native address format when its peers lie in different type transport networks that use different address formats. A key requirement of MPTN is to allow a transport user to use its existing address format, while the transport provider uses addresses that the network expects.

Most transport providers utilise an address format of the following canonical form:

<NetID.host-ID.local-address>

However, the specific length and structure of the component parts is likely to be different from one transport provider to the next, and in some cases may not actually exist.

Since addresses are simply bit strings, there is no assurance that a transport address, unique relative to its native protocol, will not be duplicated in another network. The MPTN solution to this problem uses ordered pairs called *MPTN qualified transport addresses* of the form:

<transport-protocol-type, protocol-specific-address>.

Because the specific address is unique within a protocol address space, the MPTN qualified transport address is unique in the universe.

Because all incoming MPTN connections and datagrams are directed to local addresses, only node addresses need be registered to the Address Mapper. The function of the Address Mapper is to return the node address, which taken together with the well-known local address allows you to establish the connection, or to send a datagram.

The Address Mapper-to-Access Node protocols are based upon connectionless communications. It uses the connectionless services of the TLPB to exchange ABM Commands as the user data part of MPTN Datagrams, supplying the Transport User source and destination addresses as appropriate. However, the format of the resulting control information (for example, the datagram header), and the routing mechanisms employed to forward such datagrams, are transparent to the Address Mapper. The Address Mapper must have simultaneous relations with virtually every node in the MPTN, and yet the typical activity is for an Access Node to start up, register a number of addresses, and then have very little subsequent activity with the Address Mapper except for address resolution function when it needs to find a non-native path to a potential partner. Therefore establishment and maintenance of a connection would be unnecessary overhead and could indeed swamp the node containing the Address Mapper Server. All the Address Mapper-to-Access Node flows are connectionless, and there is no ordering or *a priori* relationship between the Address Mapper requests.

Each Address Mapper request is idempotent; that is, a subsequent request to perform the same action is valid and the state changes that occur as a result are normal.

1.3 MPTN Address Mapping Services Functions

The MPTN Address Mapper provides the mapping function when neither the algorithmic technique nor the protocol-specific directories technique are used. The functions performed by the Address Mapper Server and its Clients include:

- *Initialisation*

  An Address Mapper Server is a transport user and hence it follows the normal procedure to initialise the MPTN services. When the node comes up, the address mapper identifies itself to the CMM. How this happens is implementation-dependent.

  When an Access Node starts up, it needs to find out where the Address Mapper Server is located and what the transport user address of the address mapper is. A search protocol may be available from the transport provider (on a protocol-specific basis) to broadcast a message throughout the SPTN (Single-Protocol Transport Network) to find the location of the available Address Mapper(s). If the search procedure returns the transport user address and the transport provider address of the Address Mapper, then initialisation of the Access Node is complete. If a protocol-specific search is not available, the Address Mapper Client may have a table of potentially available Address Mapper Servers, supplied by the user during configuration. The Access Node registers itself with one or more Address Mappers before it registers any transport users.

  The first command sent from the Access Node to the Address Mapper must be a Register of itself; that is, the Access Node Client must register its own transport provider addresses with the Address Mapper Server. The second command must be a Clear, to assure that any address mappings from a previous incarnation are deleted. Clear deletes all address pairs registered for the Access Node, except the mappings for the Access Node Client itself.

- *Synchronisation*

  The main purpose of Synchronisation is to verify path integrity between the Address Mapper Server and its associated Access Node Clients and to verify that the items registered at the Address Mapper are one and the same as the items believed to have been registered by the Access Node.

  The Address Mapper Server keeps a running list of all its active Clients and their transport provider addresses and CMM names on non-volatile storage. When the Address Mapper Server comes up it consults this list and sends an ‘‘I’m back’’ message to each Client. This message includes an indicator showing whether the address mappings have been preserved on a nonvolatile database, and a count of the address pairs registered at this Address Mapper Server on behalf of each client. If the address mappings have not been preserved, or if the Client perceives that the wrong number of address pairs is registered, it can initiate recovery.

  Also included in this list of Clients is a time stamp that is reset whenever a command is received from the Client.

  A particular transport user’s address mapping may become ‘‘of dubious validity at this time’’ if another resource attempts to connect to it and reports to the Address Mapper that it is unreachable. The implication is that, while the Address Mapper Server keeps a dynamic list of address mappings, network or node failures or application crashes may cause some registered users to become unavailable to all or parts of the MPTN.

  To determine the reachability of its clients, the Address Mapper Server sends an awareness signal to the Access Node. If the Acess Node responds, any ‘‘dubious’’ markers are turned off for the transport users at that Access Node.

Every two hours, the Address Mapper Server goes through its list of Clients to determine if there are any ‘‘dubious’’ entries more than two hours old. If there are, and if the Access Node is still unreachable, the Server deletes all the registered address pairs for that Access Node.

A *clear* protocol is provided for the Access Node to request the Address Mapper to remove all items so-far registered on behalf of the Access Node. This function can be requested at any time, and must be requested when the Access Node initializes. The *clear* does not delete the address mappings for the client itself, however.

- *Address Registration*

  When a new transport user comes up in an MPTN Access Node, the CMM registers its address along with its mapping to one or more transport provider addresses. Once addresses are registered with the Address Mapper, the corresponding network entities can be reached by nonnative transport users throughout the MPTN, using the address mapping supplied by the Address Mapper Server.

  The Address Mapper Client can also remove its names from the Address Mapper by the deregistration protocol either one-at-a-time or all at once.

- *Address Resolution*

  Before an MPTN user can communicate with a peer in the MPTN network, it must learn the transport provider address of that peer. But it only knows the peer by its transport user address. Therefore the Address Mapper provides the locate protocol to acquire the transport provider address(es) associated with a transport user.

  Another resolution facility is provided whereby the Address Mapper will retrieve all transport users associated with a particular transport provider address. Thus the following two-way mapping:

  transport provider address <---> transport user address

  can be provided by the Address Mapper. This satisfies a management requirement to find all the users associated with a particular transport provider address.

Chapter 2

Design Considerations

This chapter discusses some of the considerations in the design presented in this specification, including alternatives, rationale, and fundamental concepts.

2.1 Address Mapping Alternatives

The Address Mapper provides dynamic address mapping services for its users. Its services are required if transport users are reachable over a non-native transport provider that cannot use a protocol specific name mapping mechanism.

Based on this condition, if all Access Nodes use the same protocol-specific address mapping mechanism, for example, algorithmic mapping, then the services of the MPTN Address Mapper are not required.

Under the situations when the Address Mapper is not required, there are two protocol-specific alternatives that may be implemented:

- *Algorithmic Address Mapping*

  The transport provider may implement an architected conversion algorithm that produces an appropriate transport address from the corresponding user's address. When the transport user addresses are registered, they are algorithmically converted to the corresponding transport provider addresses, which may be enrolled in a protocol-specific directory. The algorithm used is specific to a transport user-provider pair. This solution only works when the transport provider's address space is flexible and larger than that of the transport user's address space.

  One example of algorithmic mapping is the mapping from IP addresses to SNA addresses.

- *Native Directory Extensions*

  In some cases, a protocol-specific directory can be enhanced to accommodate additional transport address formats. Using this alternative, all transport user addresses are enrolled in a protocol-specific directory. The directory services function may be enhanced to handle the additional transport address format(s) and to provide the transformation between the transport user addresses and the transport provider addresses of its peers.

  For example, the Domain Name Server used in IP can handle an additional domain such as SNA.

Address mapping services can be provided by either the Address Mapper or a protocol-specific address mapping scheme. While the alternative address mapping solutions may be less costly to implement than the MTPN Address Mapper, they are sufficient only in restricted configurations. The Address Mapper provides for dynamic address assignment, whereas other schemes are static (for example, the Domain Name Server used in IP). The algorithmic approach requires careful coordination of address tables in each participating access node. Furthermore, if there are several different address mapping schemes in the network, it may be more efficient and manageable to converge on use of an MPTN Address Mapper.

The usual considerations of security, convenience, and ease of use will determine whether a particular implementation will use algorithmic, native directory, or address mapper.

2.2 Connectionless Design

The Address Mapper uses the connectionless MPTN datagram service. This service is unreliable, in the sense that messages may be lost in the network with no notification to the user. Therefore reliability is built into the Address Mapper with a system of requests, responses, and replies. When the Access Node sends a request it initiates a transaction that may be satisfied by a reply or a response (see Section 4.1 on page 23). If it receives a reply, the Access Node must send a response to verify receipt.

Timers are set to determine when a response is outstanding too long, at which point the request is retransmitted. A fixed number of retries are attempted. If none is successful, the partner is declared unreachable.

All requests are idempotent, so that if the request was received more than once, it does not matter.

Another implication of the connectionless datagram service is that there is no ordering of requests and responses; requests do not necessarily arrive in the same order they were sent. The Address Mapper is not dependent on any relationship between consecutive requests. Where a request has to be correlated with its corresponding response and/or reply, they are related by a correlator in the ABM header. This correlator is guaranteed by the originator to be unique (within the Address Mapper-to-Access Node protocol pair) for all time. The same correlator appears on all ABM commands (the request, all responses, and the reply, if present) that are involved in the transaction.

In order to maintain synchronisation, the Address Mapper must finish the activity requested before it sends a final response/reply to the Access Node, and the Access Node in turn may not assume the activity has been completed until a final response/reply is sent.

The Access Node must keep a list of all outstanding transactions.

Since the Access Node always sets a timer when it sends a request to the Address Mapper it is necessary for the Address Mapper to respond in a timely fashion. However, some activities performed by the Address Mapper may be rather lengthy. Therefore the Address Mapper may send a preliminary response with a *pending* return code, advising the Access Node that the activity will probably not be completed before the timer expires in the Access Node. The Access Node then sets a long-activity timer and awaits confirmation of completion of the activity. This final confirmation is carried as a *reply* (since a response has already been issued to the original request), and the Access Node must send a response to the Address Mapper to confirm receipt of the reply.

A typical scenario for an application program in the MPTN environment is to initialise, resulting in registration of a transport user address with the Address Mapper Server, and then to receive requests for its services. After the initial registration, the transport user may have no further need to communicate with the Address Mapper Server. On the other hand, such an application could crash and the Address Mapper Server would have no knowledge that it had gone away. If, however, any resource in the MPTN attempts to connect to a crashed application, the connection will fail, and the resource will send a ‘‘not found’’ signal to the Address Mapper Server. If the Access Node containing the crashed application does not respond to a poll from the Address Mapper Server, the address mapping for the crashed application is marked as ‘‘of dubious validity at this time’’. If another resource requests mappings for the same application, the Address Mapper will respond noting that the path to the application is ‘‘dubious’’. At two hour intervals, the Address Mapper Server scans all its mappings looking for ‘‘dubious’’ entries. It again polls the Access Node for each such entry, and if the Access Node does not respond, then it deletes all the address mappings associated with that Access Node.

2.3 Error Recovery

If the Access Node determines that the Address Mapper is unreachable, there are three activities it can pursue in terms of recovery:

- *Give Up*

  The Access Node abandons the attempt to transmit to this Address Mapper:

- *Try Again Later*

  The Address Mapper Client may queue up its requests internally and wait for the Address Mapper Server to come back up and notify its clients that it is back.

- *Look for a New Address Mapper*

  The Access Node consults a directory or a name server, or looks in its internal tables to find the location of a new Address Mapper. Alternatively, the Access Node may use a protocol-specific procedure whereby the transport provider broadcasts a search message for an Address Mapper somewhere within each SPTN to which it was attached. If one is found, the Access Node can initiate registration activity with this new Address Mapper. It may, in fact, turn out to be the same Address Mapper discovered through a different transport network. This would be apparent because the Address Mapper's CMM address would be the same over both transports.

There is generally one and only one active address mapper per SPTN, unless the address mappers share a common distributed database. That database might be shared by address mappers from other SPTNs as well. If there are multiple active address mappers within an SPTN, then the Access Node can use any of them. When the Address Mapper Server comes up it notifies each Access Node whether or not it has access to a distributed database.

It is assumed that an address registered to one Address Mapper will be stored in the database and would be accessible to all address mappers sharing that database. If the Access Node detects lack of reachability to one mapper then it may go to any of the others as backup. The Address Mapper can use any distributed database, provided it matches appropriate characteristics. Some considerations for putting the addresses on a database include

1. Size of the network. If there are numerous registered address pairs such that the time to re-register them all would be prohibitive, it is best to keep them on non-volatile storage.
2. Prior acquisition. If there is already a database in use, then it might as well also be used for the addresses.
3. Importance of quick restoration of the Address Mapper Server after failure. If this is an important system consideration, then a non-volatile database could be used to preclude a time-consuming period of re-registration.
4. Multiple Address Mappers in a single SPTN. If there is a requirement for multiple Address Mapper Servers in a single SPTN, then a distributed database must be used.
5. Ratio of CMMs to transport users. If there is a one-to-one ratio, then it is probably best to keep track of them on disk, because the Address Mapper Server has to keep a list of Client CMM addresses on disk anyway. The overhead to store a single transport user per CMM does not seem to be onerous. If there are many transport users per CMM, and disk costs are a problem, they can all be kept in memory.

2.4 Addressing Relationships

The Address Mapper protocols require that the following addressing relationships be established between the Access Node and the Address Mapper (see Figure 3-1 on page 13):

- The Access Node needs to know the transport user address for the Address Mapper Server so that it can recognise when a single Server is accessed through two or more transport providers. See Figure 5-1 on page 32 for the protocol description.
- he Access Node needs to know at least one transport provider address of the Address Mapper so that it can be reached initially. This is generally provided in a static user-built table or a static protocol-specific directory such as IP's DNS unless the native transport provider has some broadcast capability. Clearly the Access Node cannot use the Address Mapper to find the transport provider address of the Address Mapper. See Section 5.1.2 on page 31 for further details.
- The Address Mapper Server needs the transport provider address(es) of the client itself. Therefore the first thing the client does is to register its own transport user/transport provider address mappings.
- The Address Mapper needs to remember the CMM address associated with each transport user that it registers, to preclude registering the same transport user name from two different nodes. See Section 5.3 on page 38 for a description of this protocol.

2.5 Group Registration

The Address Mapper handles dynamic registration of multicast groups. When the transport user wishes to join a group, and if the group does not already exist, the group is implicitly created and its transport user address is registered with the Address Mapper. If the transport provider does not provide inherent multicast, then the Address Mapper will reference the Multicast Server on subsequent requests to send datagrams to the group.

Chapter 3

Address Mapper Model and Configurations

This chapter discusses the internal models for the Address Mapper and its client Access Nodes and then portrays several different sample configurations of Address Mappers, Access Nodes, and disturbed databases.

3.1 Models

The models for the Access Node containing the Client and the Address Mapper Node containing the Server are shown in Figure 3-1.

In the MPTN model, only transport user addresses are valid above the TLPB, while only transport provider addresses are valid below the CMM. This is what address mapping is all about.

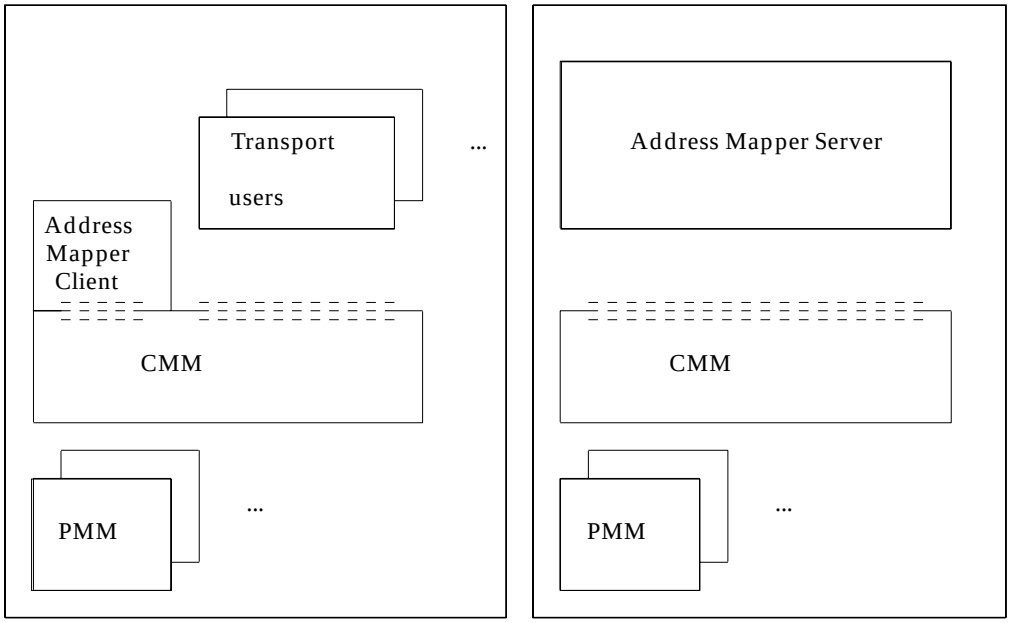

Figure 3-1 Models of the Access Node and Address Mapper Node

This illustration shows a number of transport users in an Access Node. Each of them may request MPTN Address Mapping Services through the TLPB[1] to the CMM. The CMM in turn indicates to the Address Mapper Client what is required. The Address Mapper Client is a component of the CMM that itself uses the TLPB to send requests to the Address Mapper Server. These requests are then sent just as any other MPTN data is sent.

1. The - - - - lines in the diagram indicate the TLPB.

When an Access Node starts up, the Address Mapper Client needs to find out the location of the Address Mapper Server. This may be static information such as supplied in a system configuration, or there may be a protocol-specific broadcast methodology in one or more of the PMMs whereby the node address of the Address Mapper can be acquired. See Section 5.1.2 on page 31 for more on this procedure.

When the node containing the Address Mapper comes up, the PMMs need to know that there is an Address Mapper resident, because they may need to respond to searches from other Access Nodes for the Address Mapper. This is discussed further in Section 5.1 on page 31.

Whenever the Address Mapper registers a transport user, it must remember the transport user address as the key for future references within the locate protocol. It must also remember the transport provider address; that is, the address of the PMM(s) in the access node over which the transport user can be reached. It must also remember the CMM address, in case a transport user with the same name in a different node attempts to register. See Section 5.3 on page 38 for further discussion.

3.2 Configurations

MPTN supports a variety of Address Mapper configurations, depending upon the complexity of the transport interdependencies.

There may be one Address Mapper per SPTN, a single Address Mapper may be shared by multiple SPTNs, or there may be more than one Address Mapper in an SPTN.

In each of these configurations, there are four cases that illustrate the relationships between the primary Address Mapper and its backup or partner Address Mappers, if they exist.

- If the Address Mapper fails (or the path to it breaks), there is no backup.
- If the Address Mapper fails, some other Access Node detects it and creates a *virgin* backup Address Mapper without any registered address pairs in it.
- There is a shadow Address Mapper that gets a copy from time to time of the primary Address Mapper file of registered address pairs. If the primary Address Mapper fails, the shadow Address Mapper can become the new Address Mapper and the Access Nodes can resynchronise their registered address pairs with the shadow file.
- The *old* primary Address Mapper should clear its store of registered address pairs so that any new instance starts with a blank slate. If there was a non-volatile distributed database involved, the Address Mapper would not need to clean out the store.

These four cases will be illustrated in a single SPTN configuration and then in a multiple SPTN configuration.

Figure 3-2 shows a single Address Mapper attached to a single SPTN.

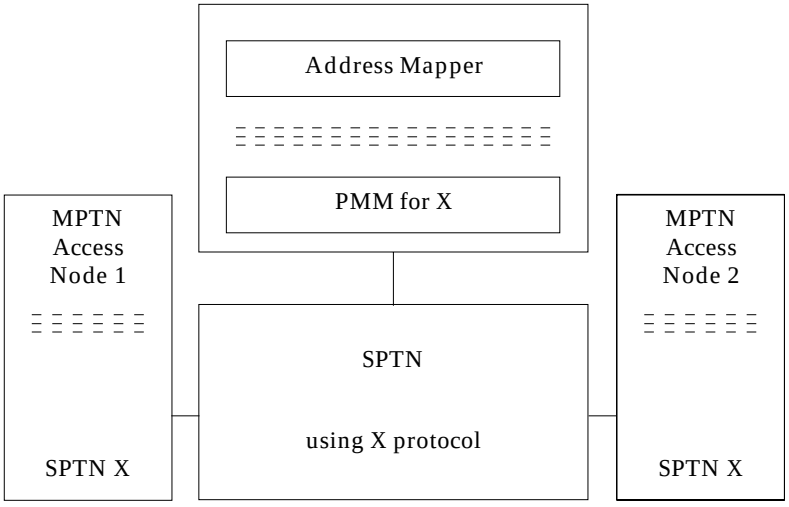

Figure 3-2 An Address Mapper in Single SPTN with no Backup

If the Address Mapper fails or a path to it is lost, there is no backup and so address mapping services are not available. This does not mean that communications across the MPTN must cease. Relationships that have already been established between transport users, based upon addressing information acquired from the MPTN Address Mapping Services when it was working, are not affected. Furthermore, if a CMM cached addressing information from the Address Mapper, it could be used to establish connections or send datagrams to its peers even after the Address Mapper failed.

In Figure 3-3, we see the same network, but Access Node 2 has the capability of creating a *virgin* backup Address Mapper. Since it has no history of previous registration activity, the Address Mapper clients have to start over from scratch.

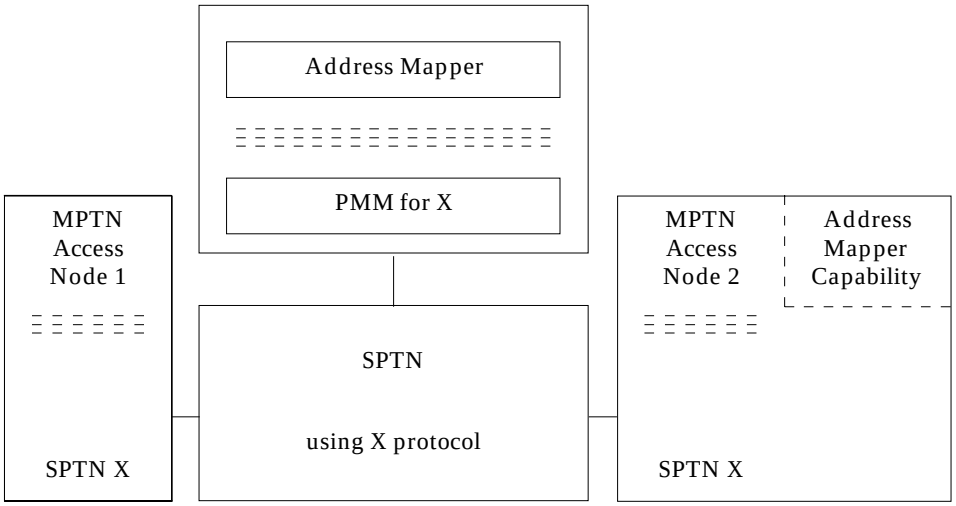

Figure 3-3 An Address Mapper in Single SPTN with Virgin Backup

The architecture does not specify the coordination between Address Mappers in a primary/backup situation. However, there are some considerations that should be incorporated in an implementation:

- Use of a *virgin* backup Address Mapper is suitable only for small to medium-sized networks.
- There should be a fairly lengthy time-out period before a backup Address Mapper assumes the role of primary.
- The backup may periodically check to see if the original Address Mapper comes back, and would then take itself down.
- The Address Mapper Server sends a message to all its known Clients whenever it comes back up. This message contains information about the Server's capabilities and is used by the Client to determine recovery actions as necessary.

The third case is shown in Figure 3-4, where there is a shadow Address Mapper in Access Node 2. The primary Address Mapper (in an implementation-dependent procedure) sends a copy of its registration files to the shadow from time to time.

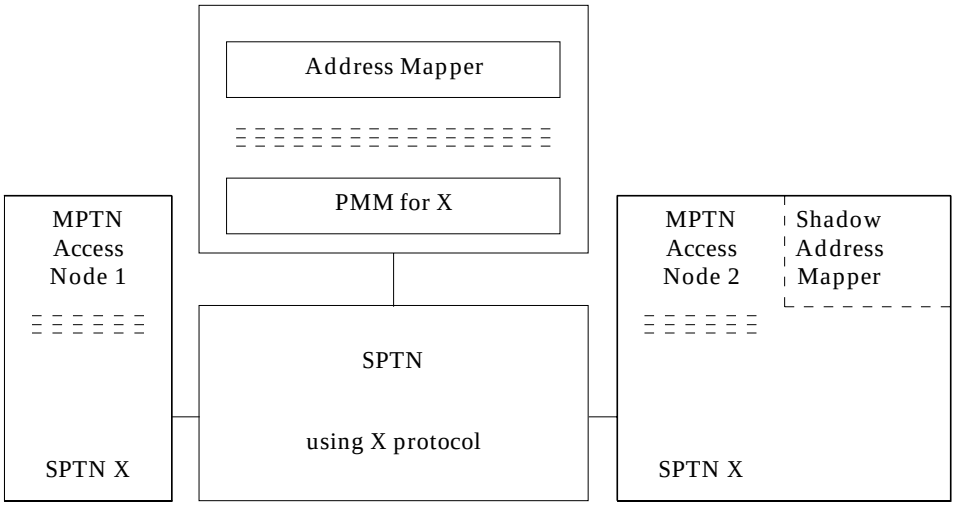

Figure 3-4 An Address Mapper in Single SPTN with Shadow Backup

If the primary Address Mapper is lost, the shadow can be used. The Access Node can ask the shadow Address Mapper for the list of addresses that it finds registered in the registration files. Based upon this information, the Access Node may leave its registration ‘‘as is’’, delete some of its registered address pairs, delete all of them and start over, and so on.

Finally, Figure 3-5 shows a primary Address Mapper with a twin that shares the distributed database.

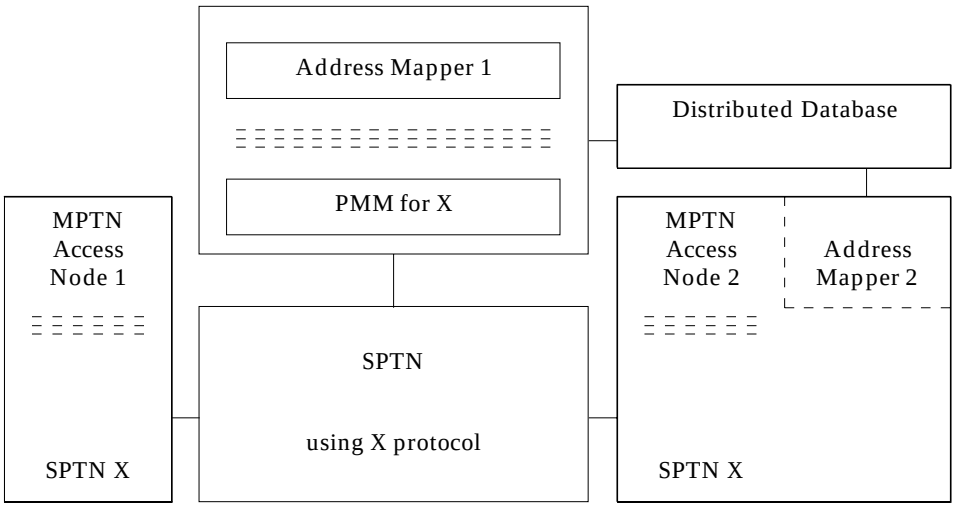

Figure 3-5 Two Address Mappers, Single SPTN, Sharing Distributed Database

When either of the Address Mappers makes a change to the database, the new information is available to both Address Mappers and so they are always in synchronisation. If the primary is lost, then the backup in Access Node 2 can be used without any explicit synchronisation activity by the Address Mapper agents. When the backup in Access Node 2 comes up, it indicates whether a distributed database is in use or not. If so, and if the count of registered address pairs is accurate, then the Clients can continue seamlessly.

If there are Address Mappers that are shared between multiple SPTNs, similar backup configurations are applicable.

In the next example, all the Access Nodes share a single Address Mapper as shown in Figure 3-6. This mapper can be reached from Access Node 1 through the X SPTN, and it can be reached from Access Node 2 through the Y SPTN.

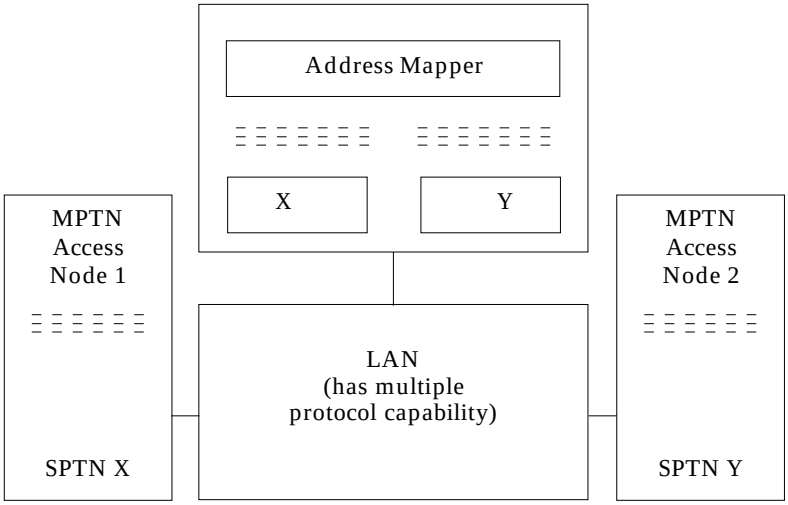

Figure 3-6 Sharing of Address Mapper by two SPTNs — No Backup

There are no backups in either SPTN X or SPTN Y, and so if the Address Mapper fails, there are no MPTN Address Mapping Services in either SPTN. As before, users in Access Node 1 can still reach users in Access Node 2, provided they have acquired the necessary address conversion information before the Address Mapper failed.

Perhaps the Address Mapper did not fail, but the path between it and Access Node 2 (that is, some problem in SPTN Y) went out. While Access Node 2 would not be able to use MPTN Address Mapping Services, the users in Access Node 1 would continue normally.

A multiprotocol subnetwork can also have multiple Address Mappers to improve fault tolerance and to distribute the traffic. If the backup Address Mapper is a *virgin* backup or a shadow similar to those shown in Figure 3-3 and Figure 3-4 above, it cannot be simultaneously active with the primary Address Mapper, but if the primary fails then the backup becomes the primary Address Mapper and each Access Node can resynchronise as necessary.

If multiple Address Mappers in a multiprotocol network use the same distributed database, they can both be simultaneously active. If either of the Address Mappers fails (or the path to it is broken), then the Access Nodes merely start using the other Address Mapper.

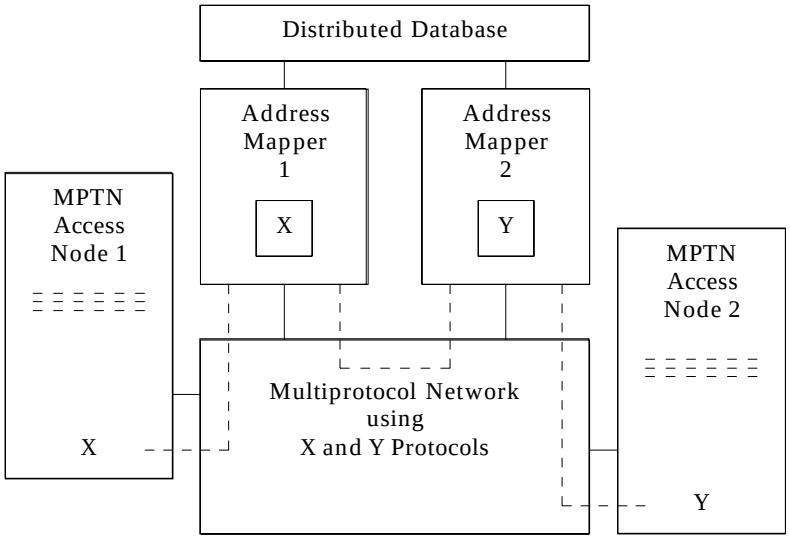

Figure 3-7 Twin Address Mappers in front of Distributed Database

Note: Either of the Address Mappers in Figure 3-7 can be used by the Access Nodes, and if one fails, the Access Nodes can begin using backup without interruption.

Generally an Access Node would find it simpler to use just one of the Address Mappers in a distributed situation such as that shown in Figure 3-7. However, there is no restriction such as that; if a big Access Node wanted to register thousands of addresses with Address Mapper 1 and immediately follow with a few thousand more registration requests to Address Mapper 2, it would work. Both sets of registered address pairs would go into the distributed database and the whole batch would be equally accessible through Address Mapper 1 or Address Mapper 2.

An Access Node may also attach to several MPTN SPTNs and thus be connected to multiple Address Mappers if these subnetworks do not share the same Address Mapper (see Figure 3-8).

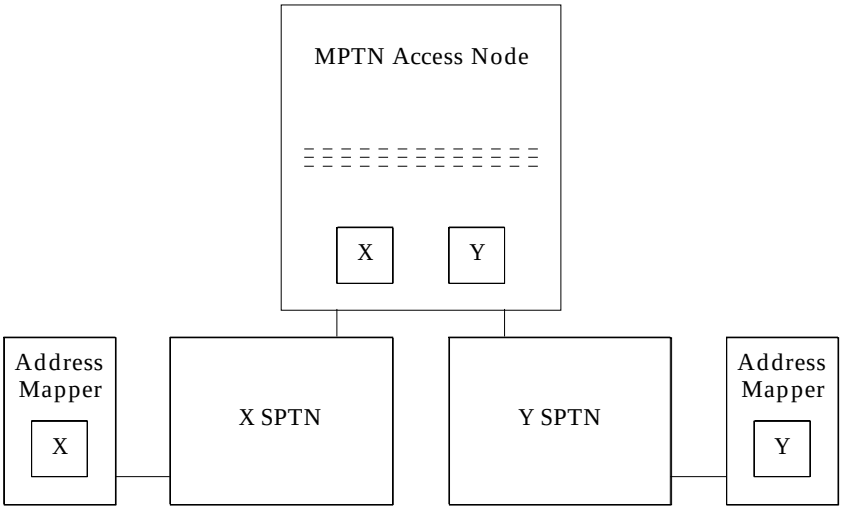

Figure 3-8 An Access Node Attached to Multiple SPTNs

Note that the MPTN Access Node attached to the X and Y networks in Figure 3-8 is not a gateway; it merely has access to both the X and Y SPTNs.

X/Open CAE Specification

Part 2:
MPTN Address Mapper Protocols

X/Open Company Ltd.

Chapter 4

Overview of Address Mapper Protocol

4.1 Address Mapper Connectionless Protocol

MPTN uses connectionless protocol between the Address Mapper and Access Nodes.

This protocol uses the MPTN datagram service to transport messages. Source and destination user addresses are carried in the datagram header, and the addresses of the Address Mapper Server and the Access Node Client are unique to facilitate routing within the MPTN components.

The datagram header contains the following fields in the datagram header source user address (SUA) and destination user address (DUA) fields:

- MPTN DG HEADER SUA from the Client to the server:

Qualifier	%	x'02' IP (or whatever transport it really is)
Mode	%	x'04' AMC Mode
Node Address	%	IP (or whatever) Node Address of the CMM
Local Address	%	IP (or whatever) Local Address of the CMM if required

- MPTN DG HEADER DUA from the client to the server

Qualifier	%	x'FF' (MPTN specific address)
Mode	%	x'05' AMS Mode
Node Address	%	x"FF' (MPTN specific address)
Local Address	%	Null

The DUA and the SUA are flipped on datagrams from the server to the client.

Note: The qualifier of the Address Mapper Server is fixed as X'FF'. The CMM uses this information and the mode of x'05' (AMS Mode) to route requests to the Address Mapper Server.

In general, an exchange between the Access Node and the Address Mapper is initiated by the Access Node with an ABM request. The Access Node sets an activity timer. and waits for a response from the Address Mapper. The default value of this timer is 30 seconds but it may be adjusted to suit the environment. When the Address Mapper receives the ABM request, it performs the requested activity to its completion and returns the appropriate ABM response, with the return code indicating successful completion or some unusual condition.

The simple case is illustrated for a Register request in Figure 4-1.

Figure 4-1 Simple Handshake with a Successful Return Code

Here the Register request is sent from the Access Node to the Address Mapper, the Address Mapper stores the user address and transport address(es), and returns the Register Response.

Note that if the Register request or the Register response were lost in the network, the Access Node would simply re-initiate the procedure by re-transmitting the Register request. If the Address Mapper had received the Register request (that is, it was the response that got lost), it would not matter. ABM requests include a transaction ID, used by the receiver to detect duplicate outstanding messages (see below).

If no response appears before the timer expires, the Access Node retransmits the request and waits another 30 seconds (default) for a response. This cycle repeats five times before the Access Node declares the Address Mapper to be unreachable.

To keep track of its active Clients, the Address Mapper Server keeps a time stamp for each Access Node. Whenever a request is received from a Access Node, the timestamp is reset. At intervals of two hours, the client list is scanned for potential garbage collection (see Section 5.2 on page 34).

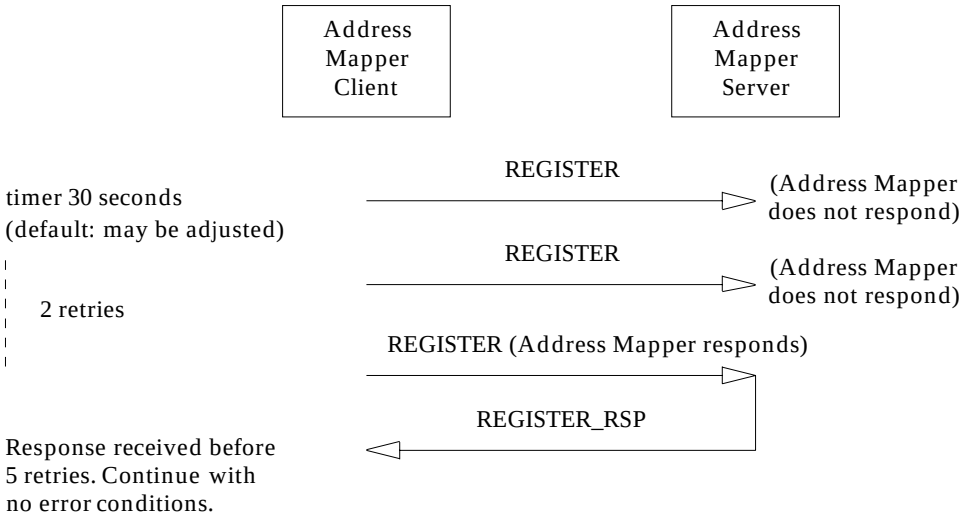

Figure 4-2 Timing out and Retransmitting Successfully

In Figure 4-2, the Access Node sends a REGISTER request and sets its 30-second (default) timer. The Address Mapper does not respond before the timer expires, and so the Access Node retransmits the Register request, using the same transaction ID. The protocol is idempotent, so

the Address Mapper can perform the same registration activity a second and third time. Of course, the Address Mapper can deduce from the transaction ID that the Access Node did not receive a response and sent the same Register request again. In Figure 4-2, after the Access Node has performed two retries, the response from the Address Mapper arrives and the transaction is concluded successfully. Had the response not been received before the fifth retry, the Access Node would have concluded that the Address Mapper, or the path to it, had failed.

There are some activities that may take a long time for the Address Mapper to complete. For example, a Locate request might involve building a very long list; so long that it is likely the activity timer in the Access Node could expire before the Address Mapper finishes building the list.

For this reason, a *pending* return code is provided on most responses from the Address Mapper Server to indicate to the Access Node that the execution of the request is likely to take a long time, and that the Access Node should reset the timer to two minutes to cover the additional activity at the Address Mapper. This timer is also user configurable to suit the environment, and if it expires, five retries are attempted.

After the requested function is complete, the results are returned in a reply. Because the reply is sent as a connectionless request, the receiver must send a confirmation response. The pending response is invalid in answer to a reply.

When the Access Node declares the Address Mapper to be unreachable it takes one of two actions. If it is a small Access Node with few registered address pairs, it may go to a different Address Mapper and issue a Clear, then re-register all its active users. Another option would be to queue the requests and wait for the Address Mapper Server to signal that it has come back up. If there is a twin Address Mapper in front of a nonvolatile distributed database the Access Node can just switch to the twin.

When the Access Node receives a response other than *pending* from the Address Mapper, the transaction is complete, as is the case when the Access Node receives a reply from the Address Mapper. If the Access Node receives a reply before a response is received, it means the pending response got lost in the network, and it should treat the reply as if the pending response had been received. If the wayward pending response comes in later, it should be discarded (see Figure 4-3).

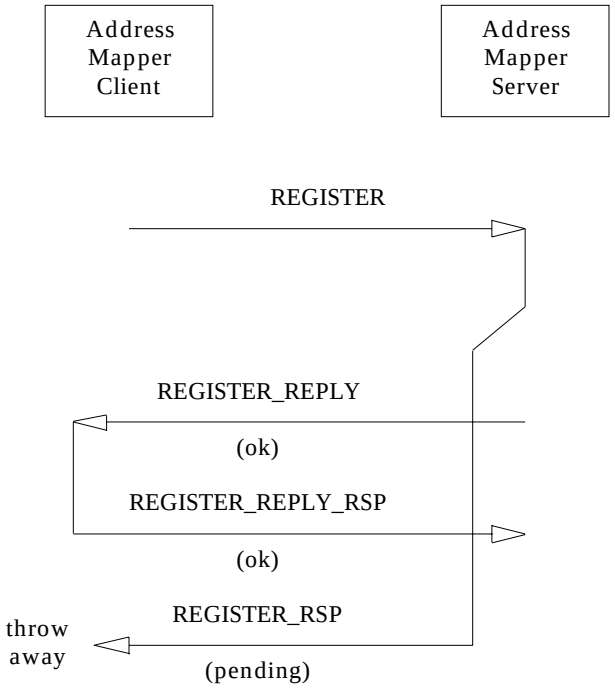

Figure 4-3 Response and Reply Returned Out of Order

In this example, the pending response took longer to make its way through the network than did the Register Reply. When the Register Reply appears at the Access Node, it can send the OK response immediately without waiting for the errant pending response. When the pending response does come in (it may be lost forever, in which case the results are identical), the Access Node simply throws it away.

A correlator in the ABM header called the transaction identifier is used to correlate requests, replies, and responses. When an ABM command is sent, the originator supplies a unique value for the transaction identifier and this value is used on all requests, responses, and/or replies that are generated as a result of the ABM command. The originator keeps a table of outstanding transactions. If a response or reply comes in with a transaction identifier that is not in the table, it should be discarded. There can be multiple outstanding (incomplete) transactions between the Access Node and the Address Mapper.

In order to facilitate lookup at the Address Mapper, a local identifier called the ABM alias address may be sent from the Address Mapper to the Access Node in the ABM header. This should be stored by the Access Node and inserted in the ABM header on requests it sends to the Address Mapper.

4.2 ABM Command Summary

The following table summarises the ABM Commands used for the Address Mapper protocols within each function. Details on each protocol can be found in the following chapters.

The following naming convention is used with the ABM Commands. The two letters in the command name that appear after the prefix ABM indicate the source and destination of the command. The following letters have specific meanings:

- *A* is the Access Node.
- *M* is the Mapper.

Thus a message that has ABM_MA as the prefix is sent from the Address Mapper to an Access Node, while a message that has ABM_AM as the prefix is sent from the Access Node to the Address Mapper.

Details on the individual messages may be found in the individual Command definitions in subsequent chapters.

All ABM Commands except R_U_THERE have responses (in particular, the Replies have responses). For ABM_R_U_THERE, a Reply is unnecessary, because the Access Node never has any conditions that cause a delay in responding.

A negative response may be sent to a request or reply. It comprises the contents of the initial command with an appropriate return code and/or suitable diagnostic information appended to the end of the response.

Table 4-1 Commands between the Access Node and the Address Mapper

ABM Functions	**Purpose**
(1) **Initialisation Function**	The procedures used when the Access Node starts up
Self-identification Procedure	The internal procedure whereby the Address Mapper identifies itself to the CMM.
Search Procedure	A broadcast method to find the location of the Address Mapper(s) reachable by the Access Node
Discover Procedure	A procedure using the destination provider address of the MPTN datagram to learn the transport user address of the Address Mapper Server.
(2) **Synchronisation**	
Enquire Protocol	Used to verify what is registered at the Address Mapper.
ABM_AM_ENQUIRE_REQUEST	The Access Node uses Enquire to find out what addresses are registered at the Address Mapper on its behalf.
ABM_MA_ENQUIRE_REPLY	Reply to the ABM_AM_ENQUIRE by the Address Mapper. Used if a pending response was previously sent to an ABM_AM_ENQUIRE. It contains a list of address pairs registered for the enquiring Access Node.
Heartbeat Protocol	Used to verify reachability
ABM_MA_R_U_THERE_REQUEST	Heartbeat from the Address Mapper to the Access Node to determine if they are mutually reachable. The Access Node always returns an OK response.

ABM Functions	**Purpose**
Clear Protocol	Used to clear out all registered address pairs for an Access Node except the mapping that points to the Access Node itself.
ABM_AM_DEREGISTER_REQUEST	The Deregister (all) can be used by the Access Node as a Clear to obliterate the registered address pairs that are retained on its behalf at the Address Mapper. Clear is always the first protocol between the Address Mapper Client and a Server.
ABM_MA_DEREGISTER_REPLY	Reply from the Address Mapper to the Access Node confirms that its addresses were deleted. It is used if a pending response was previously issued. If the request could be processed to completion within a short time, the fields defined below can be carried on the response in lieu of the Reply.
Alert Protocol	Used by the Address Mapper Server to notify its active Clients that it has just initialised.
ABM_MA_I_AM_BACK	This command informs the clients that the Address Mapper Server has come up. It indicates whether the address pairs are on volatile storage (in which case the Clients need to re-register all their transport users) or on a non-volatile database (in which case the Clients may assume that the registered transport users are intact).
Not Found Protocol	Used by any MPTN user to notify the Address Mapper Server that a connect to a registered transport user failed.
ABM_AM_NOT_FOUND	This command informs the Address Mapper Server that a registered transport user should be marked as ''of dubious validity at this time''. This could be to a transient network outage, a persistent channel fault, or a failure (crash) of the transport user itself. Subsequent LOCATEs are notified that the transport user is suspect.
(3) **Registration**	
Registration Protocol	Used to register address pairs at the Address Mapper.
ABM_AM_REGISTER_REQUEST	Registers a single transport user address, with its associated transport providers, to the Address Mapper.
ABM_MA_REGISTER_REPLY	Reply to the ABM_AM_REGISTER; confirms the registration took place. Used if a pending response was issued to a previous REGISTER.
Deregistration Protocol	Used to remove registered address pairs from the Address Mapper.
ABM_AM_DEREGISTER_REQUEST	Deregisters a single address or a collection of addresses from the Address Mapper.
ABM_MA_DEREGISTER_REPLY	Reply to the ABM_AM_DEREGISTER; confirms that the addresses have been deleted. It is used if there was a previously-issued pending response. If the request could be processed to completion within a short time, the fields defined below can be carried on the response in lieu of the Reply.

ABM Functions	**Purpose**
(4) **Resolution Function**	
Locate Protocol	Used to resolve addresses from one form to another.
ABM_AM_LOCATE_REQUEST	(1) Resolve the queried transport user address into supporting transport provider address(es) (2) Return all the transport user addresses associated with a particular transport provider address.
ABM_MA_LOCATE_REPLY	Reply to a previous ABM_AM_LOCATE_REQUEST: (1) contains the transport(s) associated with the transport user address sent on the LOCATE request (2) contains a list of transport users associated with the transport provider address sent on the LOCATE.

Chapter 5

Protocol Description

5.1 Initialisation Function

The initialisation function includes three procedures designed to initialise the relationships between the Access Node and the Address Mapper. These procedures may be invoked at the time the Access Node starts up, either at the beginning of the day or after a catastrophic failure of the Access Node. The initialisation function includes the self-identification procedure, the search procedure, and the discover procedure.

5.1.1 Self-identification

The CMM must know if there is an Address Mapper in its node. This is so that the PMMs can respond to a search for Address Mappers over their associated transport networks. The fact that there exists an Address Mapper may be hard-coded into the node or may be a system configuration parameter. Otherwise, an implementation uses internal signals between the Address Mapper and its CMM for this procedure.

5.1.2 Search

The Access Node needs to know the location of the available Address Mapper(s) in the MPTN. This may be learned by consulting a directory, performing a simple name search, or by supplying the information during system configuration. If a more dynamic method is required, the PMM may send a protocol-specific broadcast datagram through the SPTNs to which it is connected, asking for the location of Address Mappers. Depending on the transport protocol, what is returned may be a transport provider address only or a transport provider address together with a transport user address. In the former case, the Access Node would execute the discover protocol to get the transport user address of the Address Mapper. If, after waiting a long while for the response, no Address Mappers respond to the broadcast message, it must be assumed there are no Address Mappers in the SPTN.

5.1.3 Discover

Once the Access Node knows the transport provider address of the Address Mapper it uses the discover procedure to learn the transport user address of the Address Mapper Server and to make its transport provider address(es) available to the Address Mapper.

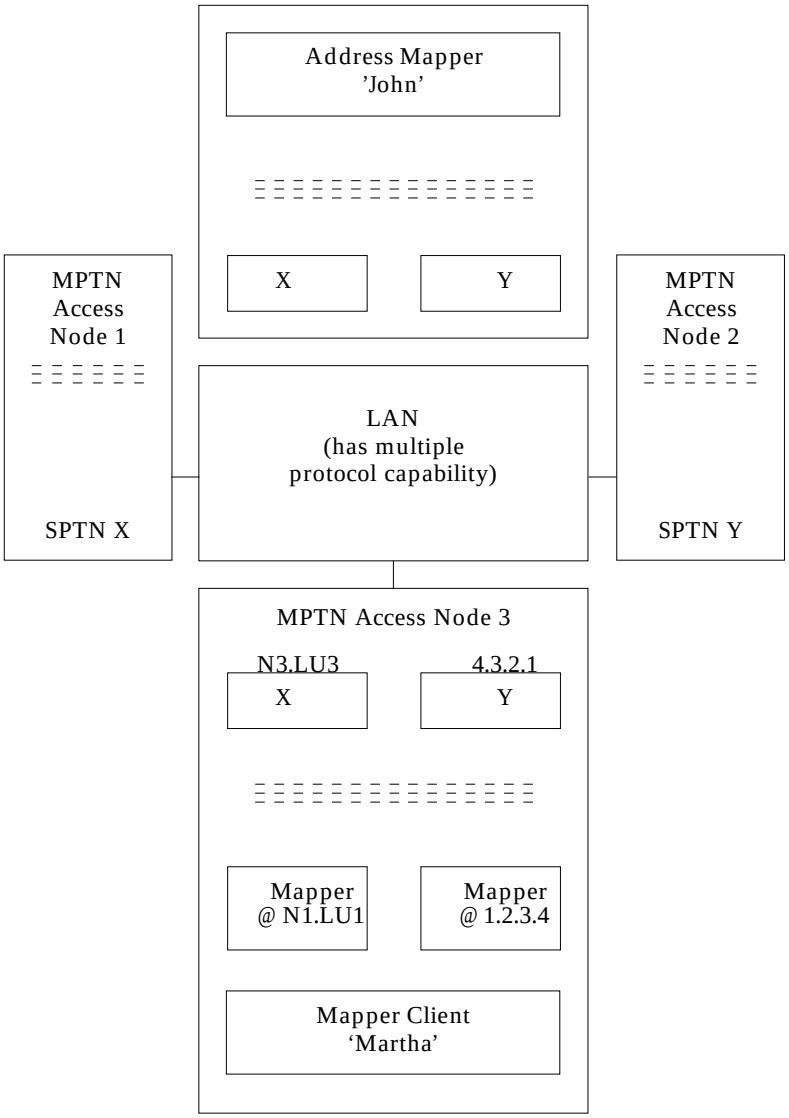

Figure 5-1 Access Node to Address Mapper via Multiple Transport N/Ws

In Figure 5-1, In this picture the Access Node has two different transport provider addresses — N1.LU1 in the X network and 1.2.3.4 in the Y network — for the same Address Mapper. It must use the discover protocol to ascertain that the Address Mapper over the X network and the Address Mapper over the Y network are in fact the same Address Mapper ''John''.

Similarly, the Access Node Client ''Martha'' has two different transport provider appearances to the Address Mapper Server. It must use the discover protocol to tell the server what transport provider addresses to use when sending ABM commands to it.

The very first exchange between the server and the client occurs when the client registers itself with the Address Mapper. This is required so that the Address Mapper Server can send responses back to the client over any of its available transport providers. In the example in the figure above, it would register itself as transport user ''Martha'' over transport provider addresses N3.LU3 in the X transport network and 4.3.2.1 in the Y transport network. The Address Mapper Server records this mapping and then uses the mapping to send the Register

response back to the client. It does not matter whether the Register travels over transport network X or Y, so long as the Register specifies both transport providers.

The datagram arrives at the CMM of the node containing the Address Mapper Server. The CMM, recognising the Qualifier as x'FF' and Mode of x'05' (AMS), passes the datagram contents up to the Address Mapper Server.

The second command sent by the client must be a Clear, in case there were address mappings left over from a previous incarnation of the client. Clearly the semantic of Clear is to clear everything except the client's own address mapping.

The last exchange between the client and the Address Mapper Server is a Clear followed by a Deregister specific of itself.

5.2 Synchronisation Function

Occasionally there may be loss of synchronisation between the Access Node and the Address Mapper. This could be because one or the other component failed or because there was an outage somewhere in the network that prevented communication between the two.

The Synchronisation Function provides the following capabilities:

- The Address Mapper Server sends an I_AM_BACK command to its active Access Nodes when it first comes up. This tells the Access Nodes if the Address Mapper has stored its address mappings on a non-volatile database, in which case the Access Nodes are unlikely to need to re-register all their transport users.
- The Access Node can request the Address Mapper to delete all registrations at the Address Mapper that are associated with itself using a form of the DEREGISTER command.
- Any component in the MPTN can notify the Address Mapper Server that an address mapping may be invalid by sending a NOT_FOUND command. If a connect fails to a resource that contains an address mapping at the Address Mapper Server, the NOT_FOUND alerts the Address Mapper to mark the transport user as ''of dubious validity at this time''. If the entry remains ''dubious'' for some time, it can be deleted by the Address Mapper Server.
- The ''dubious validity'' flag is turned off whenever the Address Mapper Server receives a REGISTER, DEREGISTER, or LOCATE command from the Access Node.
- Before the Address Mapper Server unilaterally deletes an address mapping, it sends a liveliness inquiry (R_U_THERE) to the suspect Access Node. If there is no response after the normal timeout-retry mechanism, then all the address mappings for that Access Node are deleted.

When an Access Node starts up, it must register itself with the Address Mapper. But the Access Node has no way of knowing whether it has previously established a relationship with an Address Mapper and whether the Address Mapper has already registered addresses on its behalf. Therefore, the second command sent by the Access Node is a clear, to delete all registered address pairs that may have been left around. (Clear does not delete the Access Node's own address mapping.). This protocol protects against the case where connectivity was lost between the Address Mapper and the Access Node but was not detected.

5.2.1 Clear Protocol

The Clear protocol is used by the Access Node to request the Address Mapper to deregister all entries associated with the Access Node except for the mappings pointing to the Access Node itself. It may be used when the Access Node starts up, to assure that no registered address pairs are left over from a previous instance. It may also be used when the Access Node detects or suspects an out-of-synch situation between itself and the Address Mapper where the registered address pairs are not the same set as those the Access Node wishes to have registered.

This protocol works as follows. The Access Node sends a Deregister All to the Address Mapper requesting that all registered addresses associated with the Access Node be deleted.

If this is the initial interchange with the Address Mapper, it first uses the discover procedure (described in Section 5.1.3 on page 31) to get the transport user address of the address mapper.

When the Address Mapper receives a Deregister All request, it searches the database to see if there are addresses that have been registered for the requesting Access Node. If there are none, the Address Mapper returns a Deregister Response. If there are registered address pairs for that Access Node, they are all deleted and then the Deregister Response is sent. The Address Mapper may return a preliminary *pending* response to the Deregister, followed by a Deregister Reply, as

discussed in Section 4.1 on page 23.

5.2.2 Alert Protocol

The Address Mapper Server uses the Alert Protocol to notify all its active clients whenever it comes up. Its active client list is stored on nonvolatile disk at the node, along with a count showing each client's registered address tuples (where a transport user registered with multiple transport providers counts as many tuples as there are providers registered with it). If the count disagrees with the Client's expectation of the number of tuples it believes to be registered, then it may take remedial action. It may issue a Clear and re-register all its transport users.

The I_AM_BACK also carries an indicator showing whether the Server uses a nonvolatile database or not. If it does not, then the Client is obliged to re-register all its transport users anyway. The I_AM_BACK conveys the Address Mapper Server's transport user address, which needs to be stored at the Client.

In order for the Address Mapper Server to comply with this protocol, it must keep track of all its active Client CMM names and transport user addresses on disk.

ABM_MA_I_AM_BACK

The I_AM_BACK command is sent by the Address Mapper Server to inform its active clients that it has just come up.

Since the action taken by the Access Node is immediate, there is no need for a reply form of the I_AM_BACK command.

Address Mapper's Alias (in the header)
: This is the Address Mapper's alias address. The Address Mapper Client should store this address and supply it on all subsequent requests to the Address Mapper Server.

Count of Registered Pairs
: This is the number of address pairs or transport user/provider tuples stored at the Address Mapper Server on behalf of the Address Mapper Client to which this I_AM_BACK command is being sent. A tuple is a single transport user paired with a single transport provider. If there are multiple transport providers registered with one transport user, then there are as many tuples as there are transport providers.

Volatility Flag
: This indicator shows whether the Address Mapper Server keeps its address mappings in volatile or nonvolatile store. If the mappings are in volatile storage, then the Address Mapper Client has to re-register all his transport users. If in a non-volatile database, the Client may assume that all his transport users are still registered, unless there is a discrepancy between the Count of Registered Pairs and the number of transport user/transport provider tuples the Client believes to be registered at the Address Mapper Server. If there is a discrepancy, the Client can issue Clear and then re-register all his transport users.

  If the Client has any pending Registers or Deregisters (that is, those that have been sent to the Address Mapper but have not been finished or confirmed), it should reissue all of them.

5.2.3 Not Found Protocol

When an attempt to connect to a transport user fails, and that transport user's address mapping was obtained from the Address Mapper Server, the Server must be informed with a NOT_FOUND command. The Server then attempts to contact the Access Node containing the suspect transport user with an R_U_THERE command. If it fails, the transport user is marked as ''of dubious validity at this time''. The ''dubious'' flag is reset whenever a command is received from the Access Node that contains the ''dubious'' transport user.

When a LOCATE is received for a transport user marked ''dubious'', the Address Mapper Server indicates this on the LOCATE reply. If the connect fails, then the Access Node should *not* send back a NOT_FOUND.

ABM_AM_NOT_FOUND

This command informs the Address Mapper Server that a transport user was not found, though its address mappings were reported on an earlier LOCATE reply. The Address Mapper marks the transport user/transport provider mapping as ''of dubious validity at this time'' and sends a positive response.

Since the action taken by the Access Node is immediate, there is no need for a reply form of the NOT_FOUND command.

Transport User Address
: The address of the transport user that apparently failed.

Transport Provider Address
: The Transport Provider Address that was used in the failed attempt to connect to the transport user.

5.2.4 Connectivity Verification Protocol

Every time a command is received from an Access Node, a time stamp is associated with the Access Node. Every two hours, the Address Mapper Server goes through its client list to find entries marked as ''of dubious validity at this time''.

Any MPTN resource that has a connection fail to a transport user whose address mapping was acquired from the Address Mapper Server is expected to send a NOT_FOUND command to the Server. The Address Mapper attempts to contact the Access Node containing the transport user with a R_U_THERE command. If it does not respond, then the transport user mapping is marked as ''of dubious validity at this time''.

Successful completion of an R_U_THERE protocol does not verify the existence of a particular transport user, only that the Access Node is up.

When a LOCATE is received for a transport user marked "dubious", the Address Mapper Server indicates this on the LOCATE reply. If the connect fails, then the Access Node should *not* send back a NOT_FOUND.

Whenever a command is received successfully from the Access Node, any "dubious" flags associated with transport users on this Access Node are turned off.

As the Address Mapper Server periodically goes through its Client list, it looks for mappings marked ''of dubious validity at this time''. For any such transport users, it sends an R_U_THERE to the Access Node containing them. If the Access Node does not reply (after the normal timeout-and-retry sequence), then the Address Mapper Server unilaterally purges the address tables of *all* entries associated with that Access Node.

There are no additional fields in the R_U_THERE beyond the standard headers.

If the Access Node receives the R_U_THERE command, it returns an immediate positive response.

R_U_THERE

When the Access Node receives this command it simply returns a positive response. There are no fields in the request or response.

Since the action taken by the Access Node is immediate, there is no need for a reply form of the R_U_THERE command.

5.3 Address Registration Function

The Registration function allows an Access Node to register its non-native addresses with the Address Mapper so that other Access Nodes can access the corresponding transport users. A complementary Deregistration function is provided for removing such addresses from the Address Mapper's memory.

The registration protocol uses the ABM_AM_REGISTER request. The deregistration protocol uses the ABM_AM_DEREGISTER request. Each of these requests may take a long time to complete at the Address Mapper. If this is the case, the Address Mapper returns an immediate *pending* response followed up later by the REGISTER or DEREGISTER reply. The protocol used in these *pending* situations is described in Section 4.1 on page 23.

The Address Mapper must take care to prevent duplicate registrations of transport user address. In general, there is nothing to prevent *using* the same transport user address in more than one node in an MPTN environment. But duplicate transport user addresses cannot be *registered* in the Address Mapper, or an ambiguous situation would be introduced. Therefore, the Address Mapper Server always checks the source CMM name on the Register command. If the transport user is already registered by a Client at this CMM, the registration is allowed. Thus a Client can register subsequent transport providers for a transport user that is already registered. If CMM name is different, the Address Mapper Server rejects the registration as a duplicate registration attempt. However, if the registration were from a Gateway it *would* be allowed, since a transport user can be accessed through multiple Gateways having different CMM names.

The registration may be for a specific transport user, or it may indicate a wildcard. A wildcard registration (typically used only by gateways) provides for part of the registered name to be unspecific, using a bit mask. The part of the name that is unspecific does not have to match precisely on a subsequent LOCATE request.

On a wildcard registration, the Address Mapper Server stores both the transport user address and the mask (as well as the associated transport provider address(es)). When a LOCATE comes in with a specific transport user name, the Address Mapper Server checks all the specific names first, but if there is no match, it looks through the wildcard entries. If there is a match, using the bits that are not ignored by the mask, then the Address Mapper Server notes this as a potential match. However, it must look through all the wildcard entries, note all the potential matches, and then choose the one with the longest match to send back on LOCATE. If there are multiple wildcard matches of the same length, then the LOCATE sends them all back.

The registrant's MPTN type field identifies the type of MPTN component that requested a registration function. Generally this will be an Access Node, but there are cases when certain network components are allowed to issue registration requests on behalf of other components.

If there is a user data field in the register request that the Address Mapper Server receives, it stores the information contained therein in the address tables and associates it with the transport user. If a subsequent LOCATE specifies this transport user, the Address Mapper returns this user data in the LOCATE response.

There is an optional field in the register command that can specify a user exit. If this field is present, the Address Mapper Server stores, along with the transport user and its transport provider addresses, a pointer to this user exit. If the transport user is selected on subsequent LOCATES, then this user exit is invoked before the LOCATE response would be returned. Upon return from the user exit, the information supplied is placed into the LOCATE response (actually it is probably a reply since this could take a frightfully long time) and sent back to the requesting Client.

The Load Level field is used by the Registrant to convey certain information about its current load level, or how busy it is at the moment. The information is stored by the Address Mapper

along with all the other attributes.

The registration procedure for groups is similar. The ABM_AM_REGISTER command contains the transport user group address and a null value for the transport provider address. If the group already exists, the address mapper returns the transport provider group address and increments a reference count for the group. For each transport provider to which it has access, the Access Node uses the transport provider group address to register with the multicast group, using either the Multicast Server (MCS) protocols or the transport provider's inherent group join protocol. The Access Node ignores transport provider addresses for providers that it does not have.

If the group does not exist and the transport provider has inherent multicast capability, the Address Mapper attempts to create/join a transport provider group. If successful, the Address Mapper registers the group, sets the reference count to 1, and returns the transport provider group address.

If the group does not exist, and the transport provider does not have inherent multicast capability, the Address Mapper returns the transport provider address for the Multicast Server. The Access Node then joins the group using the MCS protocols. The Multicast Server then registers the multicast group with the Address Mapper, and the Address Mapper sets the reference count to 1.

5.3.1 Registration Protocol

The registration protocol is used by the Access Node to register a single transport user address with the Address Mapper. It may be reachable through multiple transport providers, and the list of transport providers supplied on the Register command is assumed to be in preference order of the transport user that is being registered. The order is derived from economy, efficiency, or other preference criteria.

Registration occurs when the transport user issues an M_BIND_DC with the indication that the address should be registered with the MPTN address mapper. The list of transport providers is derived from information on the M_CREATE_DC.

The CMM address is associated with the registered transport user address by the Address Mapper so that the identity of the registering node can be determined. A duplicate address is discovered when a transport user address has been registered with different CMM addresses.

If multiple gateways register the same wildcard, the Address Mapper Server should register the transport provider addresses indicated for each of the gateways.

If the user exit field is present on the register request, then the user exit must be set up in the address tables so that subsequent LOCATEs for this transport user will cause the user exit to be invoked.

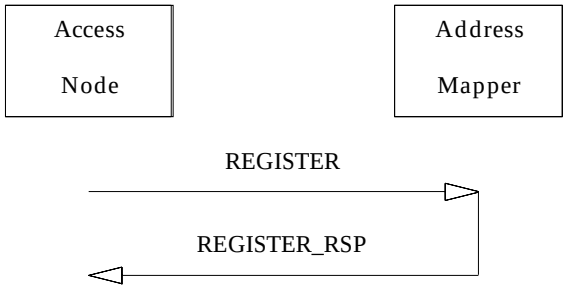

Figure 5-2 Simple Registration of a Unique Address

If the registration process is expected to take an unusually long time (such that the timer at the Access Node is likely to expire before the Address Mapper can send the response), then a pending signal must be sent to the Access Node. The Access Node resets the timer to a longer value and the Address Mapper then sends the reply when it finishes the registration task.

ABM_AM_REGISTER

The Register request contains a single transport user address followed by a list of one or more transport providers.

Information is set in the Address Mapper Header Command Modifier to indicate if the transport user name refers to a multicast group.

If the registration request is for a group, then the Address Mapper executes the group registration procedures described in the previous section.

Upon the receipt of a REGISTER command, the Address Mapper Server should reset the time stamp associated with the Access Node that sent it. The Server should also turn off the "dubious" flag of any transport user associated with the Access Node.

Verification Flag
: Set on if verification is required. If the addresses are unique, then this flag is turned off.

Transport User Address
: This is the address of the entity to be registered at the Address Mapper. If it is a group address, this is indicated in the Mode field.

Transport Provider Addresses
: A list of one or more transport provider addresses.

User Data Field
: Optional information pertinent to the MPTN user that is to be stored with the transport user's address mappings. This user data is returned on a LOCATE reply when directed to to this transport user.

Address Mask
: An optional bit mask indicating which parts of the transport user address are to be used when matching with subsequent LOCATE commands. ''One'' bits (b'1') correspond to bits that should be included in the comparison, while ''zero'' bits (b'0') correspond to bits that should be ignored in the comparison.

Registrant's MPTN Type
: Indicates the type of MPTN resource that initiated the Registration protocol. Used if a component registers on behalf of another component.

Load Level
: This field shows the load level, how busy the registrant is at the moment. It is stored with the other attributes. The load level is computed as the ratio of the number of current sessions to the maximum number of sessions that can be supported at the node, normalised to a scale of 256:

  So, if:

  s = current number of sessions supported at the node
  m = maximum number of sessions the node can support

  the value to report is:

  loadlevel = (s / m)*256.

  Once the number is reported, it should only be reported again if the number changes by a value greater than 32.

  If this optional field is not present, the Address Mapper should store the median value (128).

Limited Use Cache Count Field
: This optional field is set by the registrant to indicate that the address mappings associated with this transport user are not permanently valid. This value is the maximum number of times a CMM should use this transport/provider mapping before clearing the cache and issuing another Locate to get an up-to-date mapping.

ABM_MA_REGISTER_REPLY

The reply to the REGISTER request indicating success or failure.

If the request could be processed to completion within a short time, the fields defined below can be carried on the response in lieu of the Reply.

Return_Code
: The Return_Code conveys the reason whether the registration succeeded or it failed.

  OK
  : The registration was successful

  Duplicate Address
  : The transport user address was already registered by another Access Node.

  Address Conflict
  : Multiple existing registrations exist for the same address. This could happen if a user failed to verify a previous registration.

  Conflict with Individual Address
  : The registration was not performed because the transport user group address supplied by the Access Node conflicted with an individual address that was already registered with the Address Mapper.

  Multicast Group Already Exists
  : The transport user group address supplied by the Access Node was already registered as a group address with the Address Mapper. Therefore the Address Mapper returns the transport provider address(es) that it has registered. The Access Node uses these addresses to perform group joins over the respective transport providers to which it has access.

5.3.2 Deregister Protocol

The deregistration function is the reverse of registration; it is used by the Access Node to request the Address Mapper to delete one or more registered address pairs from its memory. There are four types of Deregistration, each of which may be used in the appropriate situation:

- All registered address pairs associated with this Access Node.

  This is used in the clear protocol to delete from the Address Mapper every address pair associated with a particular Access Node. The Access Node would typically use this form of deregistration just before shutting down or if there were a serious lack of synchronisation with the Address Mapper. The Address Mapper Client is required to send a Clear every time it comes up, to assure that nothing is left over in the address mapping tables from a previous incarnation. If the CMM in the Client Node detects that a transport user has failed (crashed or abended), then the CMM should issue a Deregister for that transport user.

- All transport users associated with a single transport provider address.

  If an Access Node needs to shut down one of its protocol stacks, it sends this form of deregistration to delete all registered address pairs that use that transport provider.

- Single transport user, all associated transport providers.

  The Access Node would use this form of deregistration when a transport user was preparing to shut down and did not want to respond to any subsequent communication from its peers.

  Deregistration occurs when the transport user issues an M_UNBIND_DC and the MPTN address mapper was used in the registration process.

- Single transport user, Single transport provider address.

  This form of deregistration would be used if a transport user or application wanted to prevent being accessed through a particular transport provider.

If a transport user requests deregistration of a non-registered transport user address, the CMM will detect it and will not forward it to the Address Mapper. If the Address Mapper is not reachable when a transport user attempts to deregister a name, the Common MPTN Manager will flag the deregistration request and send the ABM_AM_DEREGISTER after the Address Mapper is reachable.

Deregistration of a group address is similar to deregistration of an individual address, except that the Address Mapper must decrement the reference count for the group. When the reference count goes to zero, the registered address pair is deleted.

The response can be *pending* or anything described in the reply.

ABM_AM_DEREGISTER

The Deregister request specifies which type of deregistration function is required, and supplies the appropriate address to the deregistration function.

Upon the receipt of a DEREGISTER command, the Address Mapper Server should reset the time stamp associated with the Access Node that sent it. The Server should also turn off the ‘‘dubious’’ flag of any transport user associated with the Access Node.

Deregister Type

00 Clear all registered address pairs for this Access Node. This form must be sent as part of Access Node initialisation.

01 Delete all transport users associated with this transport provider

10 Delete this transport user and all its associated transport providers

11 Delete this transport user with a particular transport provider

Transport User Address
This is the address of the entity to be deregistered at the Address Mapper. This field is not present when the Deregister Type is 00 or 01.

Transport Provider Address
The transport provider address. This field is not present when the Deregister Type flag is 10 or 00.

ABM_MA_DEREGISTER_REPLY

The Reply to the DEREGISTER. If the request could be processed to completion within a short time, the fields defined below can be carried on the response in lieu of the Reply.

Return Code
This field contains the result of the deregistration request.

OK
Successful Completion

Unauthorised Action
The Access Node requested to deregister items that were registered by another

5.4 Address Resolution Function

Address resolution is required when a user makes a request to send a datagram to, or set up a connection with, a partner. This section discusses the situation where the MPTN Address Mapper performs the resolution.

It is also possible to resolve a transport provider to all its associated transport users, to satisfy the requirements of network management. Therefore a form is provided where a transport provider address is included on the request and the response from the Address Mapper indicates all the transport users that use that transport protocol address.

The Access Node uses ABM_AM_LOCATE_REQUEST to request resolution services of the Address Mapper. It may be that the Address Mapper requires a relatively long time to process the command, in which case it may return a *pending* response to the LOCATE, followed up later by an ABM_MA_LOCATE_REPLY that contains the desired results. This *pending* protocol is discussed further in Section 4.1 on page 23.

The LOCATE command includes the requestor's transport user's address. In some cases, this information may be valuable to certain auxiliary address resolution services.

5.4.1 Locate Protocol

There are three complementary functions supported by the locate protocol. First, given a transport user address, the Address Mapper returns a list of one or more transport provider addresses by which that transport user can be accessed. Second, given a transport provider address, the Address Mapper returns all the transport users that are associated with that provider. Third, given a particular MPTN type, the Address Mapper returns registered address pairs registered by that type of MPTN component, relative to the indicated mask.

- Locate Protocol for User-to-Providers (Type 1)

  Given a destination's transport user address, the Access Node uses an ABM_AM_LOCATE to request the Address Mapper to return the transport provider addresses corresponding to the given transport user address.

  The following cases can occur:

  1. The address is found.

     The Address Mapper returns the transport provider address(es) to the Access Node.

  2. The address is not found.

     In this case, a negative response indicating ''user not found'' is returned to the Access Node.

  3. A wildcard entry matches the specified transport user address. In this case, there may be more than one valid wildcard entry so the Address Mapper Server must scan all the wildcard entries and then reply using the one with the longest match. If there are more than one with the same length match, then the Address Mapper Server returns all of them. They may be from different CMM's; this is acceptable when a wildcard has been registered. it just means the transport user may be accessible through more than one gateway.

     If there is more than one transport user matching a wildcard search, their appearance in the Locate reply is dependent upon the load level parameter associated with each transport user, from low to high.

     A type 1 Locate occurs when a transport user issues an M_CONNECT_DC or an M_SEND_DG_DC across the TLPB and the CMM uses the Address Mapper for address

resolution. If, however, the address resolution information has been cached, the Locate need not occur.

- Locate Protocol for Provider-to-Users (Type 2)

  The Locate Protocol can be also used given a transport provider address, to return all the transport users that are associated with that transport provider address.

  This version of Locate satisfies a management requirement to be able to efficiently find all the users of a particular transport. It is best to get that information from the address mapper, because you can get it all with one communication, and because the manager might not have connectivity to all the Access Nodes. From an implementation standpoint, if you do it once in the Address Mapper, then it is done, otherwise you would have to replicate it in all the Access Nodes.

- Locate Protocol for Types (Type 3)

  This type locate protocol can be used to request all the registered addresses that were registered by a particular type MPTN component. For example, the requestor may wish all the SNA addresses that were registered by an MPTN gateway.

  If more than one transport user satisfies the Locate parameters, the Address Mapper uses the load level parameter to sort the transport users in the Locate reply, from low to high.

ABM_AM_LOCATE

There are three scenarios:

1. The Locate request is used by an Address Mapper Client to ascertain the location and transport provider address of an entity it believes to be somewhere in the MPTN. It supplies the transport user address to the Address Mapper and (if successful) gets back one or more transport provider addresses supporting the transport user.
2. The Locate request is used to retrieve the transport user addresses of all the users associated with a particular transport provider address. The Access Node supplies a transport provider address to the Address Mapper and (if successful) gets back one or more transport user addresses that use that provider.
3. The Locate request is used to find all registered address pairs that were registered by a particular MPTN component type. The Access Node indicates what type it wants and the Address Mapper returns a list of registered address pairs that were registered by that MPTN type. The wildcard mask can be used in this situation to show whether the requestor is looking for *all* such addresses or a subset based upon a portion of the transport address.

Upon the receipt of a LOCATE command, the Address Mapper Server should reset the time stamp associated with the Access Node that sent it. The Server should also turn off the ‘‘dubious’’ flag of any transport user associated with the Access Node.

Type Locate (in the header)
: Indicator showing whether the Access Node is requesting type 1 (user-to-providers), type 2 (provider-to-users), or type 3 (component type) Locate

Service Mode (Types 1, 2, & 3 Locate)
: This is the service mode that may be specified for the communication between the source and the destination. The Address Mapper does not act directly on the Service Mode parameter, but may pass this information along to an additional component in some MPTN environments. A positive response will be returned if the destination can be reached with one of the specified service modes. If nothing is specified, then any Mode is acceptable.

Requestor's User Address (Types 1, 2 & 3 Locate)
: The transport user address of the MPTN component that issued the Locate request. This may be used by some auxiliary address mapping components when the identity of the session initiator is critical to the assignment of a session partner, as for example in parallel session support.

Transport User Address (Type 1 & Type 3 Locate)
: This is the requested transport user address. For a Type 1 Locate, the Address Mapper returns a list of one or more transport provider addresses associated with the given transport user address.

  If it is a Type 3 Locate, the Address Mapper finds all the transport users that were registered by the type of MPTN component specified on the Locate request. It returns a list of the transport users, each with a list of its respective providers.

Transport Provider Address (Type 2 Locate)
: This is the transport provider address for which the Address Mapper is to return all transport users that are registered with this as one of their transport providers.

MPTN Type (Type 3 Locate)
: The MPTN Type, if present, indicates that the Address Mapper is to return only transport users that were registered by this type of MPTN component. The MPTN type was supplied as an optional field on Register, and if it was present, was stored in the address mapper tables with the transport user. This field is most useful in conjunction with the mask field, so that a number of addresses may be returned.

Wildcard Mask (Type 3 Locate)
: This field is an optional bit mask indicating which parts of the transport user address should be compared with the stored addresses. ''One'' bits (b'1') correspond to bits that should be included in the comparison, while ''zero'' bits (b'0') correspond to bits that should be ignored in the comparison.

Requestor's Supported Providers List (Type 1 Locate)
: A list of transport providers types supported by the Requestor.

ABM_MA_LOCATE_REPLY

This is the reply to the LOCATE. If the request could be processed to completion within a short time, the fields defined below can be carried on the response in lieu of the Reply.

Type Locate
: Indicator showing whether the Address Mapper is returning type 1 (what transports are available to the specified user) or type 2 (which users are associated with the specified transport) Locate_Reply

Limited Use Cache Count Field (Type 1 Locate)
: This field is used in conjunction with an address cache at the Client, where resolved address mappings can be stored locally for reuse without additional Locates to the Address Mapper. If the address mappings are returned in order, depending on load level, the ordering is valid for a limited period of time. This count field tells the Client how many times it can use the cached mappings before it should send another Locate.

Transport Provider's Address List (Types 1 & 3 Locate)
: This is a list of one or more transport provider addresses that were stored in the address mapper tables as providers for the transport user address that was specified on the Locate request.

Dubious Validity Flag (Type 1 Locate)
: Indicator showing that the address mapping is suspect because another user returned a NOT_FOUND for the transport user. This could be because the transport user has failed (crashed) or its node has failed, or it could be a transient failure of the network or a part of the network. In any case the sender of LOCATE may attempt to connect to the transport user. If the connect fails, it should *not* send another NOT_FOUND.

User Data Field (Types 1, 2, & 3 Locate)
: If the user data field is associated with the transport user in the address mapping tables, it is returned in this field in the LOCATE_REPLY.

Transport User's Address Mask (Type 1 Locate)
: This entry was located by matching a wildcard with the bits as indicated in the mask. It may or may not be accessible through the gateway indicated by the transport provider address. In any case the sender of LOCATE may attempt to connect to the transport user. If the connect fails, it should *not* send a NOT_FOUND. There may be multiple paths, each registered by a different gateway having different CMM names. This is not an error situation in the case of wildcard processing.

Transport User Address List (Types 2 & 3 Locate)
: The transport user addresses that are associated with the transport provider address that was contained in the Locate request.

Return Code
: This field contains the result of the address resolution request. The request may have been successful or failed because the transport user address was not found or because it was not reachable.

  OK
  : Request was successful.

  User_not_found
  : The transport user could not be found.

  User_not_reachable
  : There was no path to the transport user satisfying the specified service mode.

  Address Conflict
  : Multiple existing registrations exist for the same address. This could happen if a user failed to verify a previous registration.

X/Open CAE Specification

Part 3:
Supplementary Information

X/Open Company Ltd.

Appendix A

Protocol Flow Notation

The following diagram shows the symbols used in the flow diagrams presented in this specification.

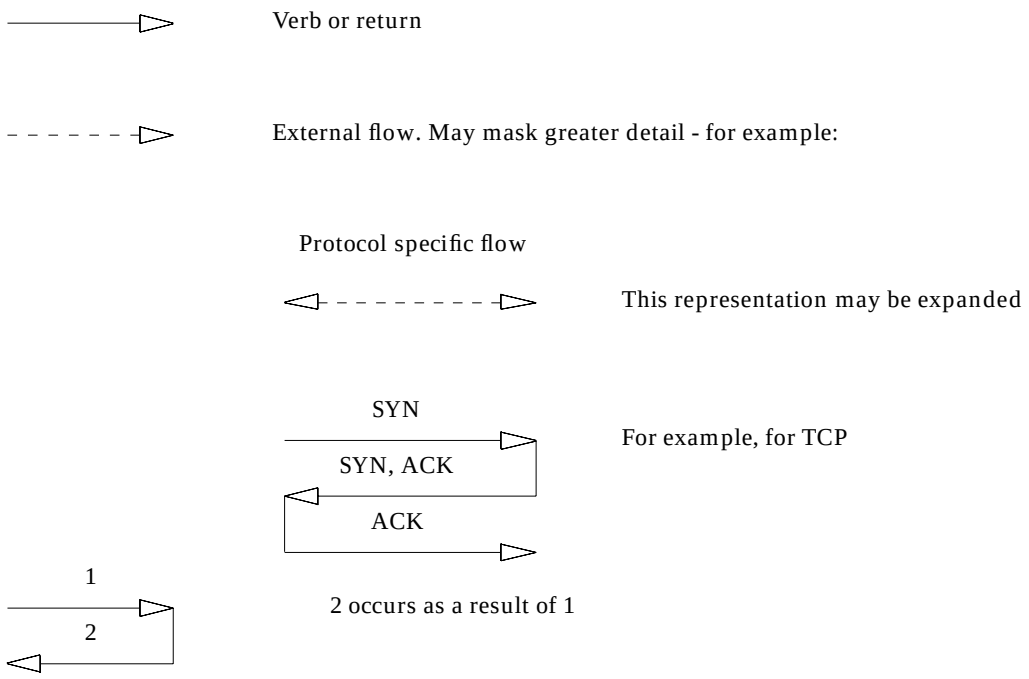

Figure A-1 Notation Used in Flow Diagrams

Glossary

The following MPTN definitions of terms and abbreviations are common to the *Multiprotocol Transport Networking (MPTN)* environment. Not all of them are used in this document.

address-mapper function
An MPTN component that maps non-native transport-user addresses to a form used in the native transport network.

address space
The set of all legal transport addresses that may be formed according to the rules of a given address type. These rules include the maximum number of characters that can be in the address and the permissible characters. Each protocol has it own set of rules. Since addresses in one protocol may also be legitimate in another protocol, MTPN qualifies all transport addresses with an address type.

address type
An identifier in an MPTN header that indicates the protocol category (for example, OSI or TCP/IP) and hence the specific syntax and structure of the accompanying address. A given transport-user address plus its address type form an MPTN-qualified address.

API
Application programming interface.

application programming interface
An interface between the application program and the application support layer.

ASCII
American National Standard Code for Information Interchange.

below-specific protocol boundary (BSPB)
The interface between the common MPTN manager (CMM) and the protocol-specific MPTN manager (PMM).

below-specific
Specific to one transport provider that exists below the CMM.

BSD
Berkeley software distribution.

BSPB
Below-specific protocol boundary.

CMIP
Common management information protocol.

CMM
Common MPTN manager.

common MPTN manager (CMM)
The component of the MPTN architecture that provides services independent of any protocol. Examples include registering transport users with the MPTN address mapper component, selecting a transport provider, and establishing MPTN connections.

compensation
The function of making up for differences in functions requested by the transport user and those provided by the transport provider.

connectionless service
A service that treats each packet or datagram as a separate entity that contains the source address and destination address. Connectionless services are on a best-effort basis and do not guarantee reliable or in-sequence delivery.

connection-oriented service
A service that establishes a logical connection between two partners for the duration that they want to communicate. Data transfer takes place in a reliable, sequenced manner.

CPI-C
Common Programming Interface for Communications.

datagram
A self-contained packet, independent of other packets, that carries information sufficient for routing from the source transport user to the destination transport user.

datagram segment
A part of a datagram. A datagram may be segmented (that is, split into more than one part) if it contains too many bytes of data to send at one time.

EBCDIC
Extended binary-coded decimal interchange code.

expedited data
Data that is considered urgent. Such data may be delivered ahead of normal data that preceded it.

group address
A single transport address identifying a collection of users. The collection of users is formed so that they can all receive common multicast datagrams.

IP
The networking protocol that forms part of the Internet Protocol suite referred to as TCP/IP. The internet protocol defines the internet datagram as the unit of information passed across the internet, and provides the basis for the internet connectionless, best-effort packet-delivery service.

LU
Logical unit.

LU 6.2
An SNA logical unit that supports general communication between programs in a distributed processing environment.

matching
The relationship between peer transport users or peer transport providers that are of the same family.

MPTN
Multiprotocol Transport Networking.

MPTN access node
A node that has MPTN components installed, allowing transport users to use non-native transport providers.

MPTN-qualified transport address
A transport address that is qualified by its corresponding address type. The address conforms to the syntax and meaning of the specified address type. An example of an MPTN-qualified transport address is the pair: (SNA, Net ID.LUname).

multicast
A technique that allows a single packet (or datagram) to be passed to a selected group of destinations that share a group address.

multicast datagram
A packet that is sent to more than one partner.

multiprotocol node
An implementation that supports more than one transport protocol.

multiprotocol transport networking (MPTN)
The architecture for mixed-protocol networking.

native
A relationship between a transport user and a transport provider that are based on the same transport protocol.

native network
With respect to a particular transport user, a transport network that provides the address type and transport characteristics assumed in the design of the transport user. No MPTN address mapping or compensation protocols are used for data transfer.

native node
A node with no MPTN capability.

native transport address
A transport-user address having the address type that corresponds to the type employed by the transport network underlying the transport user, for example, an SNA name that is being registered within an SNA network.

NetBEUI
NetBIOS Extended User Interface.

NetBIOS
Network Basic Input/Output System.

NetBIOS extended user interface
NetBEUI: the API to the NetBIOS transport protocol.

net ID
Network Identifier. The address qualifier of a transport address that identifies a group of nodes according to the network in which they reside.

In an MPTN environment, the transport user and transport provider have separate NetId domains.

Network Basic Input/Output System
NetBIOS: a protocol used by many small computers for network access.

networking
Providing a relaying and routing service.

networking protocol
A specification of the rules governing the exchange of information between components of a network.

non-native
A relationship between a transport user and transport provider that are based on different transport protocols.

non-native network
With respect to a particular transport user, a transport network whose addressing structure and transport service are different from that assumed in the design of that transport user.

non-native protocols
Protocols used by a non-native network.

non-native transport address
A transport-user address having an address type different from the transport network address type, for example, an OSI address for the target of a connection request carried on an SNA transport network.

OSI
Open Systems Interconnection. The interconnection of open systems in accordance with the hierarchical arrangement of the seven layers of networking functionality described in specific International Standards Organization standards.

PMM
Protocol-specific MPTN manager.

protocol boundary
A generic description of a functional boundary defined by the architecture; implementations must conform to the semantics of the protocol boundary, but not necessarily the syntax.

protocol-specific MPTN manager (PMM)
A component of the MPTN architecture that performs management, routing, and binding functions that are performed differently for the different transport providers.

record data format
A format that maintains record boundaries for the data being transmitted.

RFC
Request for Comment. The process by which some standards bodies define specialised solutions. In the case of MPTN, it is the definition of specialised transport protocols.

service mode
The designation by a transport user of the characteristics that must be maintained for a given connection or datagram transmission. Each networking protocol has its own way of requesting these characteristics, which must be mapped to the MPTN service mode.

single-protocol transport network
A collection of physically connected nodes that implement a single common transport protocol. A single-protocol transport network may span multiple net IDs.

SNA
Systems Network Architecture.

SNMP
Simple network management protocol.

socket
The abstraction provided by Berkeley 4.3 BSD UNIX that allows an application program to access TCP/IP protocols.

SON
Session Outage Notification - this is a transport user characteristic.

SPTN
Single-protocol transport network.

stream data format
Data that has no record boundaries. The data is simply a stream of bits.

TCP
Transmission Control Protocol.

Transmission Control Protocol (TCP)
The Internet standard transport level protocol that provides the reliable, full-duplex, stream service for TCP applications.

TCP/IP
Abbreviation for the protocols (that is, TCP, IP, UDP) that comprise the Defense Advanced Research Projects Agency (DARPA) Internet protocol standards.

TLPB
Transport-layer protocol boundary

transport characteristics
The set of transport services that a transport user expects; for example, whether data will be sent using connections or datagrams, and formatted as streams or records.

transport-layer protocol boundary (TLPB)
The MPTN protocol boundary that provides access in a protocol-independent fashion to multiple transport protocols.

transport network
An implementation of transport networking. Examples are parts of SNA, TCP/IP, OSI, IPX, NetBIOS, DECnet and Appletalk.

transport networking
The communications services provided at the transport layer and below.

transport networking protocol
A specification of the rules governing the exchange of information between components of a transport network.

transport provider
A component that provides the transport functions associated with a particular protocol stack.

transport-provider address
A transport address used to identify a transport provider.

transport user
An application program or application support element that uses transport services to convey data through a network. A program that directly requests transport services.

transport-user address
A transport address used to identify a transport user.

UDP
User Datagram Protocol.

unicast datagram
A packet that is sent to a single partner.

UNIX
An operating system originally developed by Bell Laboratories, and now owned as a trade mark by X/Open Company Limited.

User Datagram Protocol (UDP)
The TCP/IP protocol that allows an application program in one node to send a datagram to an application program in another node. UDP uses the internet protocol (IP) to deliver datagrams.

XMPTN
X/Open specification of Multiprotocol Transport Networking (MPTN).

Index