Guide

Multiprotocol Transport Networking (MPTN) Architecture Version 2

TECHNICAL GUIDES

THE Open GROUP

[This page intentionally left blank]

X/Open Guide

Multiprotocol Transport Networking (MPTN) Architecture, Version 2

X/Open Company Ltd.

X/Open Guide

Multiprotocol Transport Networking (MPTN) Architecture, Version 2

ISBN: 1-85912-116-0
X/Open Document Number: G506

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 Guide introduces the concepts and benefits of the X/Open Multiprotocol Transport Networking (XMPTN) architecture.

MPTN is a framework which allows an application associated with one networking protocol to run without change over a different networking protocol. The impact of the MPTN architecture is anticipated in the following areas:

- achievement of true transport-independence for the communications interfaces, such as XTI, sockets and CPI-C
- removal of restrictions on networks where applications can run.

The document is in two parts: a rationale, and an architecture model description.

Part 1 answers the following questions:

- What problems is MPTN intended to solve?
- Why is X/Open working on MPTN?
- In what scenarios can MPTN be applied?
- How does MPTN compare to other multiprotocol solutions?

Part 2 answers the following questions:

- What components are implied by the MPTN architecture?
- How do these components relate to existing transport interfaces, such as the X/Open Transport Interface (XTI), sockets and NetBEUI?

- What kinds of protocols are involved in MPTN?
- What compensations for protocol mismatches does MPTN provide?

As background information for the benefit of those interested, Appendix A presents a summary of concerns that have commonly arisen regarding the MPTN approach to mixed-protocol networking, along with responses to those concerns.

X/Open has published three CAE specifications for X/Open Multiprotocol Transport Networking (XMPTN). These are the XMPTN Access Node, the XMPTN Address Mapper and the XMPTN Data Formats, and they are identified in full in the **Referenced Documents** on page x. X/Open offers a package of all four XMPTN documents, in Document Set T504.

Trade Marks

Appletalk® is a registered trade mark of Apple Corporation.

DECnet® is a registered trade mark of Digital Equipment Corporation.

IPX® is a registered trade mark of Novell.

VTAM™ is a trade mark of International Business Machines Corporation.

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 Guide:

ISO 8072
: ISO 8072: 1986, Information Processing Systems — Open Systems Interconnection — Transport Service Definition (Connection-oriented).

ISO 8072/Addendum 1
: ISO 8072/Addendum 1: 1986, Information Processing Systems — Open Systems Interconnection — Transport Service Definition (Connectionless).

ISO/IEC 8073
: ISO/IEC 8073: 1987, Information Technology — Telecommunications and Information Exchange Between Systems — Open Systems Interconnection — Protocol for Providing the Connection-mode Transport Service.

RFC 1001/1002
: NetBEUI Applications Over TCP/IP.

RFC 1006
: ISO Transport Service on Top of the TCP, Version 3, May 1987, Marshall T. Rose and Dwight E. Cass, Network Working Group, Northrop Research and Technology Center.

RFC 1112
: Host Extensions for IP Multicasting, Internet Network Working Group.

RFC 1434
: Data Link Switching: Switch-to-Switch Protocol, March 1993, R. Dixon and D. Kushi, IBM Corporation.

SNA
: Systems Network Architecture: LU 6.2 Reference: Peer Protocols, Order Number SC31-6806, IBM Corporation.

TCP
: Transmission Control Protocol, RFC 793 (Defense Communication Agency, DDN Protocol Handbook, Volume II, DARPA Internet Protocols, December 1985).

UDP
: User Datagram Protocol, RFC 768 (Defense Communication Agency, DDN Protocol Handbook, Volume II, DARPA Internet Protocols, December 1985).

XTI (including NetBIOS)
: X/Open CAE Specification, September 1993, X/Open Transport Interface (XTI), Version 2, (ISBN: 1-872630-97-9, C318).

The following associated XMPTN specifications have been published by X/Open:

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

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

XMPTN Address Mapper
X/Open CAE Specification, Multiprotocol Transport Networking (MPTN): Address Mapper (ISBN: 1-85912-101-2, C520).

X/Open Guide

Part 1:

MPTN Problem Statement and Rationale

Multiprotocol Transport Networking is a general architecture for running upper layer services over non-matching transport networks.

This part of the MPTN architecture document describes the problems that MPTN is intended to solve and discusses why a general, multivendor solution is desirable.

Chapter 1

Problems Addressed by MPTN

In today's marketplace, there is a plethora of different networking protocols. There are protocols currently supported by XTI (OSI, mOSI, TCP/IP, SNA, NetBIOS) and others that are not (for example, DECnet and Appletalk). Each networking protocol has its own advantages and disadvantages in terms of network performance, network overhead, ease of network management, availability of important applications, presence or absence of multi-vendor implementations, and difficulty of configuration. Each has its supporters.

Network owners used to select a single networking protocol and stick to it. This is no longer true. Today, network owners are finding it necessary to install multiple networking protocols in their enterprise. This can be observed in enterprises supporting either multiple disjoint networks, and/or workstations or hosts with multiple protocols installed. In fact, some large corporations have ten or more networking protocols running in different parts of their enterprise.

Some reasons that network owners are finding it necessary to install multiple protocols include:

- Applications tend to be tied to a small number, usually one, of networking protocols. If the best application for a particular problem requires a networking protocol that is not installed in the enterprise, the network owner must install that new networking protocol in order to support the application.
- Some governments and industry associates are requiring international standard protocols, especially in Europe, yet organisations cannot abandon their investment in existing applications associated with other networking protocols.
- Business today requires cooperation between organisations that make independent networking decisions. For data exchange one or both partners may have to install an additional secondary networking protocol. Examples of this cooperation include:
  - — information exchange with vendors, suppliers, service providers and users — for example, motor companies exchange information with thousands of vendors and dealerships
  - — decentralisation of control of networking decisions both because corporations are decentralising and because reduced computing costs give smaller working groups freedom to make independent networking decisions
  - — mergers of independent organisations
  - — joint ventures and consortia involving independent organisations.

Having multiple networking protocols to support causes problems for both network owners and application vendors. These problems include the following:

- Each networking protocol has a cost in terms of computer memory, cycle utilisation, protocol overhead, network management overhead, address administration, and human skills. The more networking protocols run by a corporation, the higher the cost.
- For a particular corporation's business problems, network owners and users are faced with many protocols to choose from. Which protocol (or protocols) they choose to install is based on trade-offs between several different factors, including the applications available, development environments, various operating cost, current installed base, available human skills, and network management considerations.

- Corporations have invested huge amounts in application development that they cannot afford to abandon. The cost of rewriting or replacing these applications prevents them from changing network protocols.
- Application vendors that want to have broader markets for their products face extensive costs rewriting their applications over different networking protocols.

Chapter 2

Requirements

2.1 Mixed-protocol Networking

Users want to find the best solutions for their business problems. They do not want to be forced into networking protocol choices based on their application choices. Nor do they want to be prevented from acquiring the best available application because it requires a network they do not support. To them, "Open" means freedom of choice, not moving from the limitations of a proprietary protocol to the limitations of a standard one.

MPTN addresses two aspects of the problem:

- Mixed-protocol networking allows an application associated with one protocol to run without change over a different networking protocol.
- Network concatenation provides the ability to concatenate different transport protocols in order to interconnect nodes between which there is no complete path using a single transport protocol.

X/Open has chosen to address only the mixed-protocol requirements, but with a solution compatible with the MPTN solution for network concatenation.

Requirements for mixed-protocol networking fall in the following categories:

- Users need access to applications and distributed services that are not available over their installed networking protocols.

  Examples include RDA over TCP, CPI-C over TCP/IP, and OSF DCE over SNA.
- Users want to reduce their network operating costs. When simplifying the network via the selection of a reduced number of protocols (preferably one), cost can be reduced by concentrating skills, reducing network management and protocol overheads, reducing address administration, reducing procedures, and reducing hardware.
- Application vendors and large corporations need to develop applications for many different network types without duplicate development expenses. They therefore want a truly transport-independent interface to networks, one that provides the transport services that they require. An interface that provides only the services available in all networks is likely to be too restrictive for their purposes.

MPTN, as a transport solution, does not address all problems of heterogeneous networking. It specifically does not address the problem of incompatible application environments.

The format, structure and meaning of the data is not handled by the different transports that may be found in existing systems. The Internet Protocol Suite, for instance, leaves all these issues to its various applications (SMTP, FTP...).

Common issues to be solved that need more than a transport solution include:

- Different presentation services:
  - different byte order
  - different character sets representations (for example, ASCII, EBCDIC, local character sets)
  - floating point representation.
- Numerous applications providing similar but incompatible services; for example, electronic mail: SMTP, X.400, PROFS, DISOSS, SNA / DS.
- Different security systems.

Application gateways can address the presentation issue — as could a server application in a client / server environment, or the Upper Layers of OSI with their self-defining data.

2.2 General Solution

MPTN is independent of the networking protocol involved, and thus offers a general solution to mixed-protocol networking. X/Open supports a general solution to mixed-protocol networking for the following reasons:

- Users need help migrating to open systems. This help includes recognising and supporting their enormous investments in application development for other protocols. X/Open wishes to develop the open systems market by demonstrating how to support these applications in TCP or OSI networks.

- X/Open has already defined an interface to networks — the X/Open Transport Interface (XTI) — and has shown how this interface maps to five different network types: OSI transport, minimum OSI functionality (mOSI), TCP/UDP transport, SNA transport, and NetBIOS. Thus X/Open has already defined an interface that could be used as the basis of a general mixed-protocol solution for different kinds of transport users and transport providers.

  However, true transport independence can only be achieved today if the application using XTI does not make any assumptions about the transport it uses. Appendix C of the **XTI** specification warns the application writer which concepts should not be expected to be supported or have the same semantics over all the different protocols. In other words, XTI on its own cannot guarantee true transport independence unless the application writer is restricted to the least common denominator subset that is supported by all protocols.

XTI is not the answer for all user mixed-protocol problems. Most existing applications do not use XTI, and it is unlikely that they will be rewritten to use XTI, since the cost of doing so would be carried by the users themselves. However the MPTN protocols specified by X/Open can be used by other networking packages as well. For example, there are packages, or subsystems, that support higher layer functions, such as APPC, RPC, or OSI TP. These subsystems can use the specified MPTN protocols to compensate for mismatches. When they do, the users that use the subsystems can run their applications on additional network types without changing their applications. Thus the protocols developed to support a transport-independent XTI can also be used to achieve transport independence for other networking subsystems.

2.3 Transport Solution

As indicated by the number of networking standards, the range of protocol services involved in network communication is very large. Choosing to address this by creating mixed-protocol stacks at the transport layer is sensible for the following reasons:

- The transport layer is the natural boundary between users and networks, since everything above is a partner-to-partner protocol, assuming either reliable connections or datagram delivery.

- Some existing network protocols, such as TCP/IP and NetBIOS, have natural boundaries at the transport layer, making it an easy place to establish an interface between users and networks.

- There are great similarities in the transport functions provided by different protocol stacks, making it practical both to define general services and to specify the mappings between two protocols. This is the highest point in the protocol stack where this is true, and thus, the point where mixed stacks can be created with the least compensation and the least redundancy.

  Since the overlap is so great, the problem of compensating for differences can be reduced from a many-to-many problem to a few-to-few problem. Thus, instead of having distinct ways to provide each protocol combination, such as OSI-to-TCP, OSI-to-NetBIOS, OSI-to-SNA, SNA-to-OSI, and so on, we can define a set of functional compensations that can be used for identified mismatches wherever they occur. Thus the orderly-over-abortive-close mismatch can be compensated for in the same way, rather than one way for TCP-over-OSI, another for SNA-over-OSI, and so on. This set of defined compensations can be used as a toolkit from which a particular protocol combination can be built. Thus to run SNA applications over OSI transport, we first identify which SNA features are not provided by OSI, and then put together the set of compensations required to make the match. The commonality between protocols is thus exploited to standardise a small number of compensation modules, including the definition of the necessary compensation protocols and formats which would cover all of the probable combinations of transport user over non-matching transport provider.

  Prototyping activity has shown most of these compensations to be simple, and involves adding at most a label and a length to the data provided by the transport user before sending it over the transport provider. For example, record over stream compensation involves inserting a length field at the beginning of each record. Orderly over abortive compensation involves exchanging a signal to guarantee that all data in transit has been sent/received. Once the signal has been exchanged, the abortive close can be issued without losing any data.

  A few compensations are more complex — for example, providing expedited data over a transport provider that does not support expedited data natively. This compensation involves sending the expedited data as an out-of-band, acknowledged datagram to avoid being blocked by pacing constraints. In addition, the expedited data is also sent as in-band data, since datagrams are unreliable.

- The transport layer is often well below the application interfaces seen by users. When this is so, it is possible to change the network protocol by making changes to the intervening system programs, leaving application programs unchanged.

  Since there are considerably more applications than systems, the total cost to the enterprise of this solution of the application interoperability problem should be significantly lower than rewriting applications to run over standard interfaces.

2.4 Migration to OSI

Users express interest in two different approaches to migration to OSI:

- Running OSI applications and higher layer services over other networking protocols. OSI has many excellent applications which are not available on other protocols. However, users can not afford to install and maintain an OSI network just to be able to use these applications and services. MPTN will allow these OSI applications and higher layer services to run on users' existing networks, without installing the entire OSI network.

  In the case of the transport provider being TCP/IP, the method used is the Internet Standard RFC 1006, which is also in the process of becoming an ISO Standard.

  Examples of OSI applications and higher layer services include X.400 mail, X.500 Directory Services, FTAM, RDA, and OSI TP.

- Replacing other networking protocols with OSI layers 1 through 4, and continuing to run existing applications as well as new OSI applications.

  The industry can not encourage users to migrate to OSI without providing a way to protect their current investment in existing applications. By allowing existing applications to run, unchanged, over an OSI network, MPTN makes migration to OSI both easier and, as far as the applications are concerned, unnoticeable.

MPTN facilitates both approaches to OSI migration.

For the former, it offers a way to move OSI applications written for an OSI transport layer to another network without change. The advantages of upper layer OSI stacks, such as service-level negotiation, are preserved, even though the underlying network is not OSI.

For the latter, MPTN provides the protocols required to compensate for differences between OSI transport and the services expected by the applications. For example, OSI transport connections are abortively terminated, meaning that data in transit is not guaranteed delivery before the connection goes down. In OSI, this makes sense because orderly termination is provided by the session layer. However, some applications assume a transport network that provides orderly termination. MPTN provides simple compensations to provide this service without requiring all the other services of OSI upper layers that are not needed by the application.

2.5 Non-native Transport Users on a Single Network

The scenarios in this section contain a few cases to illustrate the requirements for a transport-user-oriented transport interface. The examples are not exhaustive; other combinations of transport users and transport providers can be substituted in each example. Figure 2-1 and Figure 2-2 on page 11 show user organisations that have chosen a single transport protocol for all communications needs. The upper layer communication service providers (TCP / IP sockets, DECnet Task-to-Task, and LU 6.2 in the examples) are carried over the single transport (OSI or TCP / IP in the examples).

Without the ability to run different upper layer communications services over a single transport provider (represented in the pictures as the Standard Transport Interface), separate routing networks would have to be set up and operated. In some cases, even separate physical networks would be required. With the Standard Transport Interface, only one logical routing network of the user organisation's choice would be required, reducing the problems of network engineering (configuration and performance management) and operation (fault monitoring, security, and accounting) to those associated with a single provider.

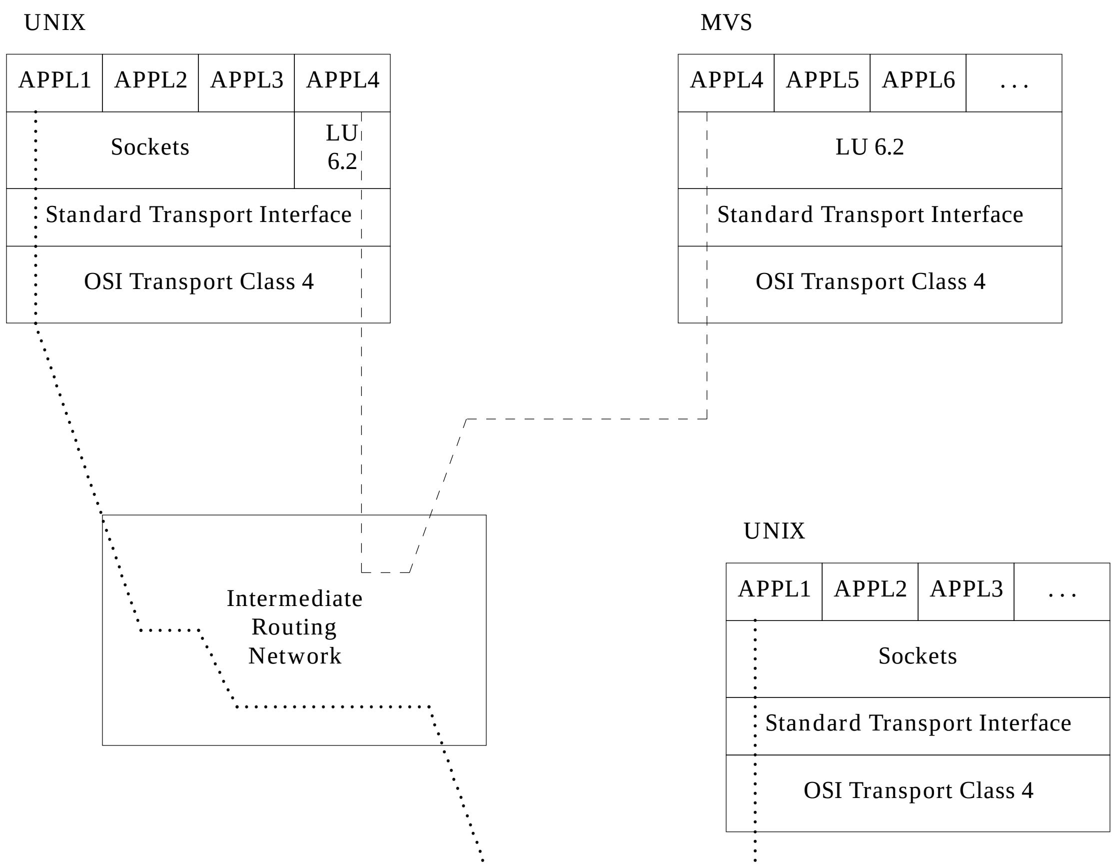

Figure 2-1 Multiple Transport User Types over OSI Transport

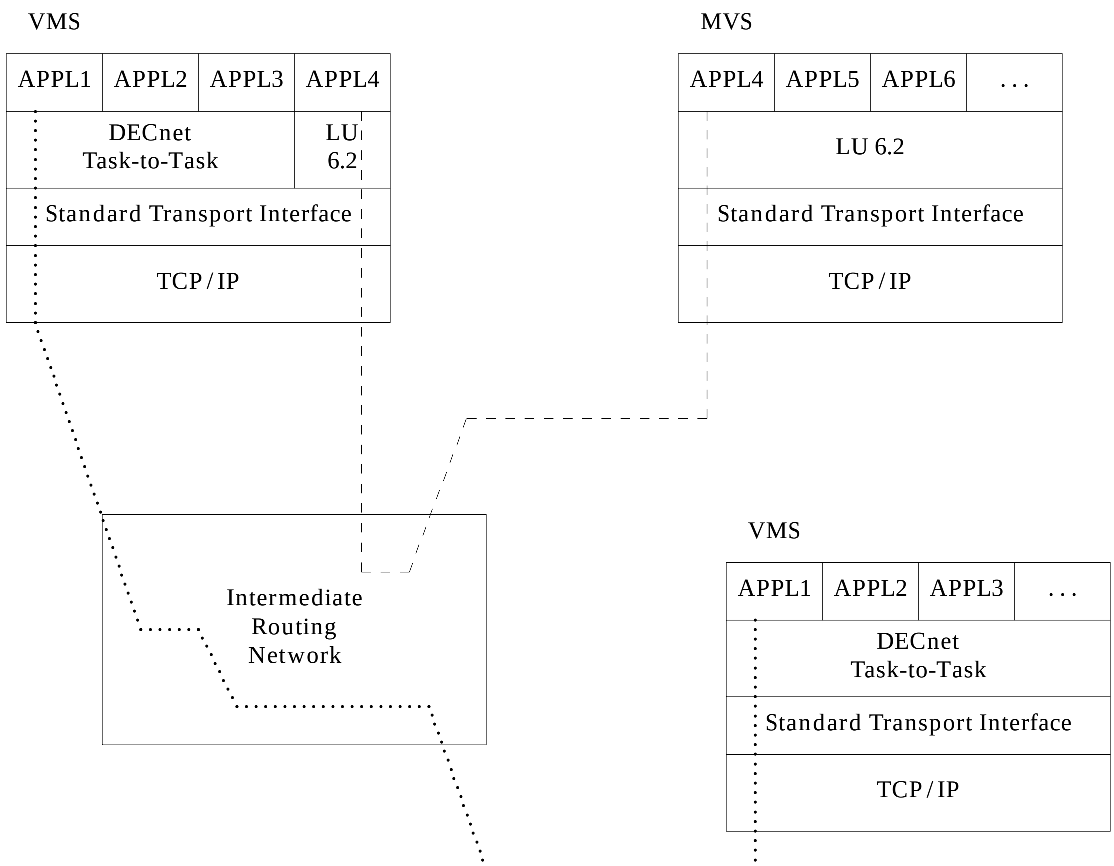

Figure 2-2 Multiple Transport User Types over TCP/IP

Chapter 3

MPTN Compared to other Multiprotocol Solutions

This section compares and contrasts MPTN to other approaches to multiple protocols that have industry backing. These approaches address the problem at all layers of networking. At the transport layer are other compensation techniques; above the transport are application gateways, middleware and remote APIs; and below the transport are multiprotocol routers and filtered encapsulation. Although MPTN is a broader, more general approach, it is not the only answer to multiprotocol networking, and has, as well as its alternatives, advantages and drawbacks. The other approaches also have environments where they are the logical choice.

3.1 Multiprotocol Nodes

One solution is simply to add each needed protocol stack to every node that needs it. This approach is used by many users today. This approach has several problems, for example, network management.

3.2 Other Compensation Techniques

Since many people have seen a need to run applications over non-native networks, specific compensation approaches have been created for particular user-provider pairs. In particular, RFC 1006 describes a way to run OSI applications over TCP/IP, while RFCs 1001 and 1002 describe a way to run NetBEUI applications over TCP/IP.

For the protocol combinations involved, these approaches have the distinct advantage that they have been implemented by several vendors. For that reason, they have been incorporated into the model described in this document.

3.3 Application Gateways

Application gateways can be created between applications on different protocol stacks that have similar functions but carry them out with different formats and protocols. For example, almost every networking protocol stack has its own electronic mail delivery mechanism: X.400 on OSI, SMTP on the Internet Protocol Suite, DIA and SNA/DS on SNA, and so on. Application gateways have been built to allow mail originating in SNA/DS to be delivered to a partner using X.400 or SMTP, and vice versa. The advantage of application gateways is that they allow the partners to run with no local software changes. With an MPTN approach, the SNA/DS user could install X.400 to run over its SNA transport, allowing the same application-level protocols to run end-to-end over two transport networks connected by a transport-level gateway.

Application gateways are required for each type of application that needs to be interconnected. While mail and file transfer may take care of a large quantity of user needs, there are many more specific applications for which it is not practical to build application gateways.

For very common applications such as mail delivery, existing application gateways may provide perfectly adequate service between networks. The continued existence of multiple mail delivery systems is likely to continue indefinitely, causing application gateways between them to be needed indefinitely.

However application gateways can be expensive to build and will probably not be available for all applications that users want to run over different networks.

3.4 Middleware

Middleware is a layer of software between the user application, and the transport network. Middleware presents an application-specific interface to the user, and applications written to this interface can run over many different transport protocols. Middleware often provides its own layer 5 and 6 functions for things like acknowledgements, commit, roll back, and presentation services.

While this makes applications written to this interface independent of the underlying transport protocols, it does not support existing applications. Existing applications need to be rewritten to use this new API. This requires that users learn a new API.

An example of middleware would be a distributed data base product. It has it's own API, and applications writen to the API can run over several different transport protocols (for example, TCP/IP, SNA, DECnet, AppleTalk).

From a vendor's point of view, this approach increases the cost of system development and requires considerable investment on networking issues, rather than on enhancing product functionality.

MPTN could be thought of as a middleware solution. However, there are two significant differences between MPTN and middleware:

- MPTN does not present a new application layer API. Thus, MPTN supports existing applications with no changes to those applications.
- MPTN is at the top of layer 4 - the transport layer. MPTN does not duplicate layer 5 and 6 functions, since those functions sit above MPTN.

3.5 Remote APIs

Remote APIs are usually found in workstations that want to access remote services via a specific API, but do not want to run the protocol and maintain the state associated with the API. The workstation uses its basic protocol, such as NetBIOS or Appletalk, to send the API call text to a server. The server carries out the real protocol associated with the API calls, returning the results to the workstation. Thus the workstation is provided with the same function as a local API user of the server protocol machine.

With remote APIs, the workstation is limited to the function available at the API, and thus may lose some underlying function such as automatic commitment of local data during a two-phase commit operation. Also, two workstations both using the API cannot communicate directly without the intervention of an API server in the middle. In addition, the server can potentially become a performance bottleneck.

3.6 Multiprotocol Routers/Filtered Encapsulation

Multiprotocol routers implement the lower-layer routing protocols from two or more protocol stacks on the same hardware sharing networking facilities, such as router-to-router links.

Filtered encapsulation approaches allow the formats for one protocol to be sent across another as user data; when the encapsulated formats exit the foreign protocol, the protocol associated with them is executed between the end points. An example is data link switching, for which IBM has published a draft RFC (RFC 1434). This technique allows SNA to use a TCP/IP network as a link; the SNA formats are encapsulated in TCP/IP messages. Because undifferentiated forwarding of traffic across a wide-area network can cause severe performance problems, all modern encapsulation techniques perform filtering. Filtering significantly increases the cost of this technique because it requires understanding of the traffic being carried.

One of the main advantages of these two approaches is that they are widely available, from several different vendors, and support many different protocols. However, they do not solve the problems caused by having multiple protocols installed in workstations and hosts.

X/Open Guide

Part 2:

MPTN Architecture Model

MPTN access architecture includes the formats and protocols that allow two transport users to interoperate over a single, non-nonnative transport network. The goals of MPTN access architecture are:

- to allow applications associated with any networking protocol to run over non-native transport providers without changes to either the application or the transport provider
- to take advantage of the similarities that exist between various transport providers in order to keep the solution as simple as possible
- to have a structure that makes it as simple as possible to serve new application types or to add new transport providers.

This part of the document provides an overview of MPTN access architecture, identifies the various components of the model, and describes the ways in which they interact.

The reference model does not go into detailed consideration of the various components, but addresses the general properties of components within the model, their means of interaction, and the properties of their interfaces.

Certain fundamental definitions used in describing this MPTN Architecture Model are supplied in the Glossary at the end of this Guide.

Chapter 4

MPTN Access Architecture Overview

MPTN access architecture is new functionality interposed between transport users and non-native transport providers in order to compensate for the differences so that both can run without change. MPTN access architecture provides formats and protocols to carry out these functions.

MPTN function appears in two types of nodes, as shown in Figure 4-1. The two different kinds of MPTN function, labeled MPTN-1 and MPTN-2, are described after the diagram.

- *MPTN Access Nodes*

  MPTN access nodes allow applications to run on non-native transport providers. These nodes may contain an interface like XTI that allows programs to be written directly to transport services, or they may contain higher layer network access services that are themselves users of transport services, or both.

  With MPTN access architecture, the two interoperating MPTN access nodes are attached to the same transport network.

  A set of architected MPTN Service Modes are defined by which a TLPB user can specify the quality of the connection or the datagram transmission that is required. The Service_Mode parameter on certain TLPB verbs passes this information to the CMM. It is passed as a single name (not a set of values) that requests, on an end to end basis, a specified level of:

  — security

  — delay

  — cost

  — capacity / throughput.

  For connection using a non-native transport, the transport user will map its method of requesting level of service to the correct MPTN Service Mode. The PMM then maps the MPTN Service Mode to the transport provider's method of requesting level of service.

- *MPTN Address Mapper Nodes*

  An MPTN address mapper node is an MPTN access node with an additional function - the address mapper - which maintains a data base of mappings between transport user addresses and transport provider addresses. This function serves other access nodes running on the same transport network. The address mapper uses the available transport services to communicate with other MPTN access nodes, and is thus a transport user itself.

  The general address mapper function is only required if the available transport providers do not include a native mechanism or algorithmic mapping for resolving transport user addresses to transport provider addresses.

Figure 4-1 shows two access nodes communicating over a single, non-native transport provider network. The address mapper shown in this picture may or may not be needed, depending on the transport provider capabilities. Thus a TCP / IP transport provider can provide address mapping via the Domain Name Server, while an OSI transport provider has no mechanism for supporting non-native addresses, and therefore requires the services of a general address mapper.

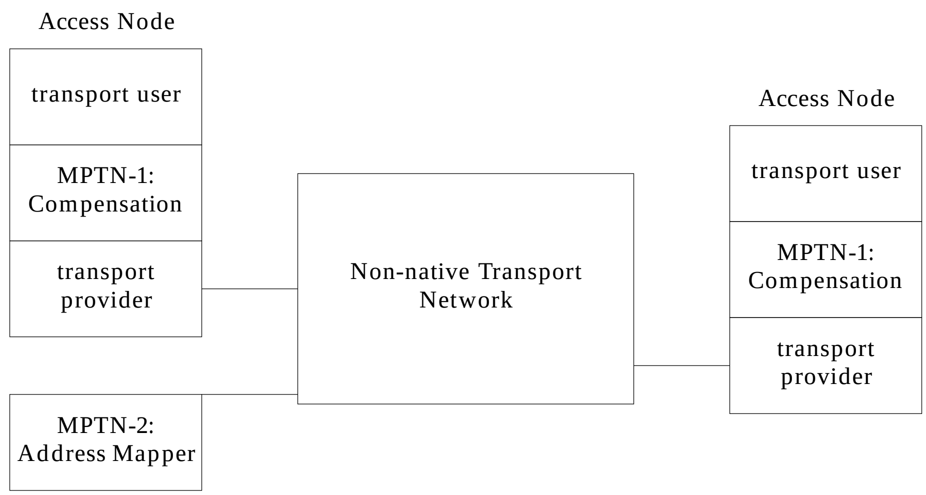

Figure 4-1 MPTN Functions in a Network

The MPTN functions identified by the labels MPTN-1 and MPTN-2 are:

- *Compensation in MPTN Access Nodes (MPTN-1)*

  The compensation bridges the gap between the needs of the transport user and the services provided by the underlying transport provider.

  Different compensation packages are needed for different transport providers serving the same transport user.

- *MPTN Address Mapping (MPTN-2)*

  An MPTN address mapper maintains a database of mappings from transport user addresses to transport provider addresses. MPTN access nodes register user address, provider address pairs with the address mapper. When an access node requests a mapping for a partner transport user address, the address mapper returns the corresponding transport provider address.

  The database can be maintained differently by different vendors; only the protocol between the address mapper and other nodes needs to be standardised.

Chapter 5

MPTN Access Model

The diagrams in this chapter show the components that comprise the various types of MPTN nodes shown in Chapter 4. The boxes are functional components; the solid connecting lines are interfaces between them. The arrows indicate the direction in which control may flow. Data may flow in both directions across each interface.

Dotted lines indicate interactions between components that may be in different nodes. These components communicate with each other by using available transport services.

Briefly, the MPTN access architecture components are:

- *Common MPTN Manager (CMM)*

  This performs the compensations that can be done the same way for multiple transport providers, including using the MPTN address mapper to resolve addresses.

- *Protocol-specific MPTN Manager (PMM)*

  This performs any part of the compensation functions that depends on the underlying transport provider. It provides a mapping to the native interface of the transport provider. It may use a native address mapping mechanism.

- *Address Mapper*

  This compensates for addressing differences if the transport provider has no means to do so.

5.1 MPTN Access Node

Figure 5-1 is the most basic MPTN model, showing the MPTN components in an access node that allow a transport user to access non-native transport providers.

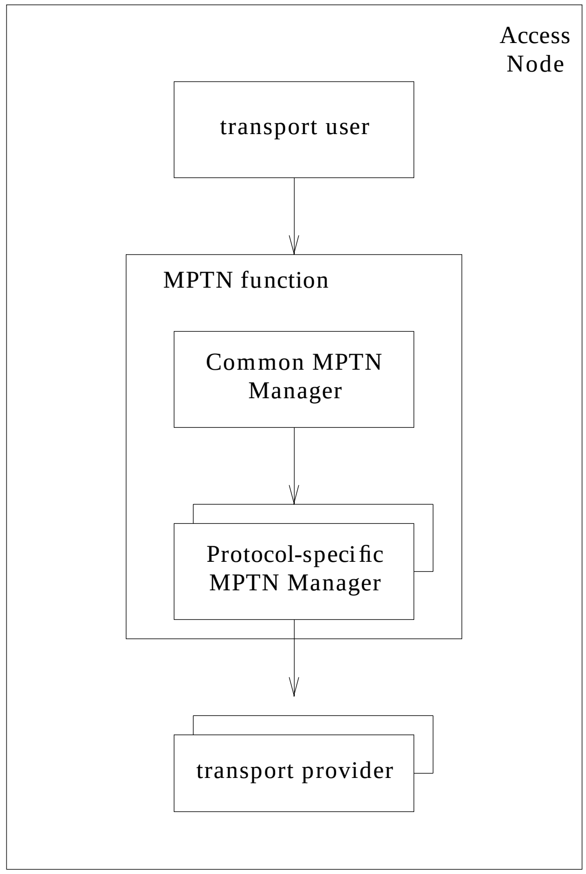

Figure 5-1 MPTN Access Node: Functional Components and Interfaces

Four components are shown in Figure 5-1, although only two are MPTN components:

- *Transport User*

  The program that requests transport services.

- *Common MPTN Manager*

  The component that connects a transport user to a particular transport provider, resolves address differences, determines which compensations are required, and performs the general compensations.

- *Protocol-specific MPTN Manager*

  The PMM is the component that performs any compensations that depend on the underlying transport provider.

  The PMM may provide a protocol-specific means of address mapping, such as the use of a TCP/IP Domain Name Server to translate from an SNA LU name to an IP address.

- *Transport Provider*

  The real provider of transport services.

Although the CMM may be involved in connection setup for native connections and datagrams to the point of establishing that the native transport provider is available and is the best match for the transport user's needs, no other MPTN services are needed for matching users and providers.

The CMM also maps from the transport user's *service mode* specification (for example, SNA class of service, OSI quality of service) to the specification supported by the transport provider. Thus a transport user can request certain services, such as encryption or fast response, that can be provided over non-native providers.

Table 5-1 summarises the salient features of SNA, NetBIOS, TCP and OSI transport providers, and shows where compensations are required for various transport services.

	SNA	**NetBIOS**	**TCP/UDP**	**OSI**
Addressing	Compensation Required	Compensation Required	Compensation Required	Compensation Required
Connection Data and Termination Data	Supported for native connections. Compensation required for non-native connections.	Compensation Required	Compensation Required	Supported for native connections. Compensation required for non-native connections.
Multicast	Compensation required	Supported	Supported, see RFC 1112	Compensation required
Expedited Data	Supported	Compensation Required	Compensation Required for true expedited data (as opposed to urgent in-line data)	Supported
Record Delivery	Supported	Supported	Compensation Required	Supported
Stream Delivery	Supported	Supported	Supported	Supported
Simplex Termination	Supported	Compensation Required	Supported	Compensation Required
Duplex Termination	Supported	Supported	Compensation Required	Supported
Orderly Termination	Supported	Compensation Required	Supported	Compensation Required
Abortive Termination	Compensation Required	Supported	Compensation Required	Supported
Session Outage Notification	Supported	Supported	Compensation Required	Supported

Note: RFC 1112, Host Extensions for IP Multicasting, is published by the Internet Network Working Group.

Table 5-1 Compensations Required over Various Transport Providers

5.2 MPTN Address Mapper

Figure 5-2 shows the MPTN components in an MPTN address mapper node and the way in which they interact with other MPTN components in other node types.

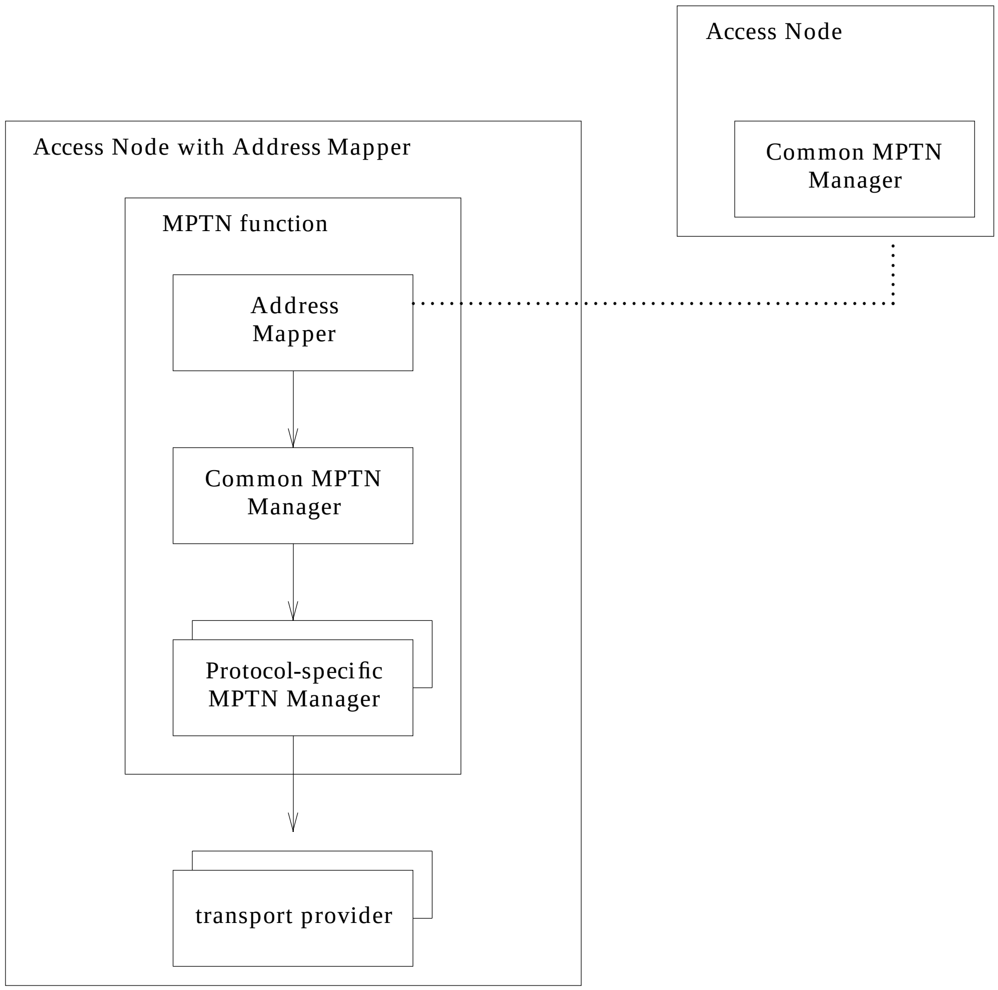

Figure 5-2 MPTN Functional Components and Interfaces (Address Mapper)

An MPTN address mapper is a transport user itself that communicates with CMMs in its own node or other MPTN access nodes. Its primary role is to resolve transport user addresses into transport provider addresses, based on the database that it maintains from address pairs registered by access nodes.

The address mapper can resolve any kind of transport user address to any kind of transport provider address. It can use any kind of transport provider to communicate with other MPTN components.

5.3 Interfaces between Functional Components

Figure 5-3 illustrates the interfaces involved in the MPTN model. The two primary interfaces are described below. Note that the existence of these interfaces is not necessary. Thus, MPTN can be used to make higher layer services run over additional transport networks without exposing an interface such as XTI for general use.

- Transport user-CMM: TLPB (Transport Layer Protocol Boundary)

  TLPB is a generic interface used to describe the semantics of the interface between the Transport User and CMM. Any transport layer interface that meets the semantics of TLPB can be used - for instance XTI, NetBEUI, and sockets. For example, MPTN can be inserted under XTI with minor syntax changes that are optional and additive. Existing XTI programs can both continue to run over XTI and also take advantage of MPTN without change.

  The primary changes are semantic changes affecting the relationship between transport user and transport provider. Without MPTN, the transport features available to the user are those native to the transport provider. With MPTN, the transport user declares the features by opening a file descriptor. The CMM determines whether compensation is required to provide the requested features. Thus, MPTN provides true transport-independence, since a transport user does not have to have different logic for different transport providers.

- PMM-Transport Provider: Native interface, such as sockets, NetBEUI or XTI

  Since the PMM is essentially an item added above the transport provider, it can be written to any interface native to the transport provider. For example, a NetBIOS PMM could be written to the NetBEUI interface, or a TCP / IP PMM could be written to the socket interface.

  Besides the functional interface, the PMM is built knowing the specific characteristics of the transport provider. Thus, part of the interface includes knowledge about the provider.

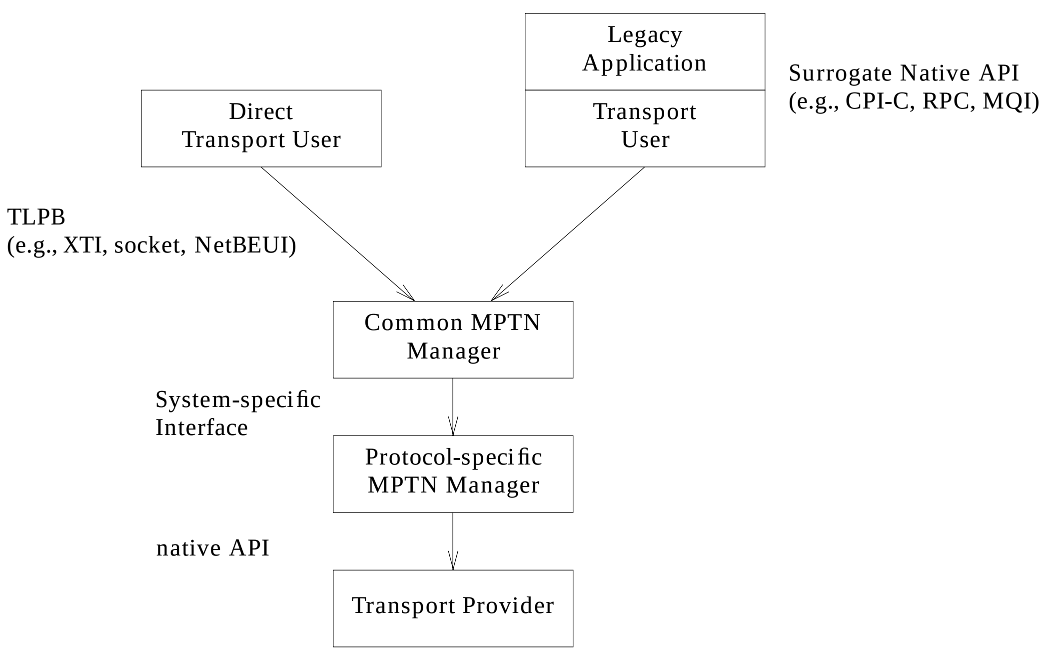

Figure 5-3 MPTN Component Interfaces

5.4 Protocols between Functional Components in Different Nodes

Figure 5-4 and Figure 5-5 show new protocols involved in MPTN access architecture. Not all protocols are required for every configuration. In fact, only the protocols shown in Figure 5-4 are required for every MPTN implementation.

5.4.1 Protocols between Two CMMs

Figure 5-4 shows the protocols between two Common MPTN Managers (CMMs) cooperating to provide MPTN function to a pair of transport users.

The primary protocols are:

- *Non-native Connection Establishment*

  In order to establish a non-native connection, the source CMM sends information on the underlying transport-provider connection including the transport user addresses of both partners, the transport characteristics required, and the compensations to be used. The destination CMM uses this information to identify the real transport user on its side and to set up the compensation mechanisms locally.

- *Non-native Datagram Routing*

  The source CMM prepends information to the user's datagram including the transport user addresses of both source and destination. The destination CMM uses this information to deliver the datagram.

- *Compensations*

  Two CMMs insert and remove information from the flow of user data on a connection in order to cooperate to give the transport users the services they require. This information is minimal, usually no more than a single octet marker and possibly a length field on a stream transport provider.

  Some compensations involve additional protocols. For example, the compensation for expedited semantics on a transport provider with no native expedited function involves sending the expedited data both as in-line data and in a datagram to an expedited server on the partner host. The version that arrives first is delivered and the other discarded. This guarantees delivery of expedited data even when the destination transport user is not receiving normal data on the connection.

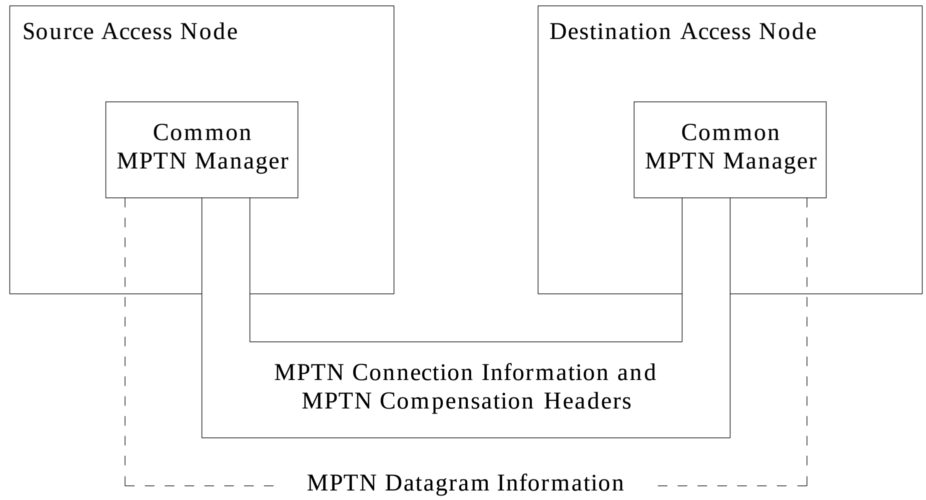

Figure 5-4 Primary MPTN Protocols (CMM-CMM)

5.4.2 Protocols between CMM and Address Mapper

Figure 5-5 shows the protocol carried out by an address mapper via datagrams with its partners.

The protocols are:

- *Registration*

  When a new transport user node address is established in an access node, the CMM registers this address to the address mapper along with its mapping to one or more transport provider addresses.

- *Resolution*

  A source CMM sends a transport user address to the address mapper requesting an address mapping so that it can bring up a transport-provider connection or transmit a datagram. The address mapper returns the destination provider address, which the CMM can optionally cache for future use.

This protocol is required only if addresses are resolved via the general MPTN mechanism. Other mechanisms, such as an algorithmic mapping or use of a protocol-specific directory, can be used for some protocol pairs. In Figure 5-5, the dotted line represents use of an existing protocol between the PMM and a protocol-specific address mapper such as the Internet Domain Name Server.

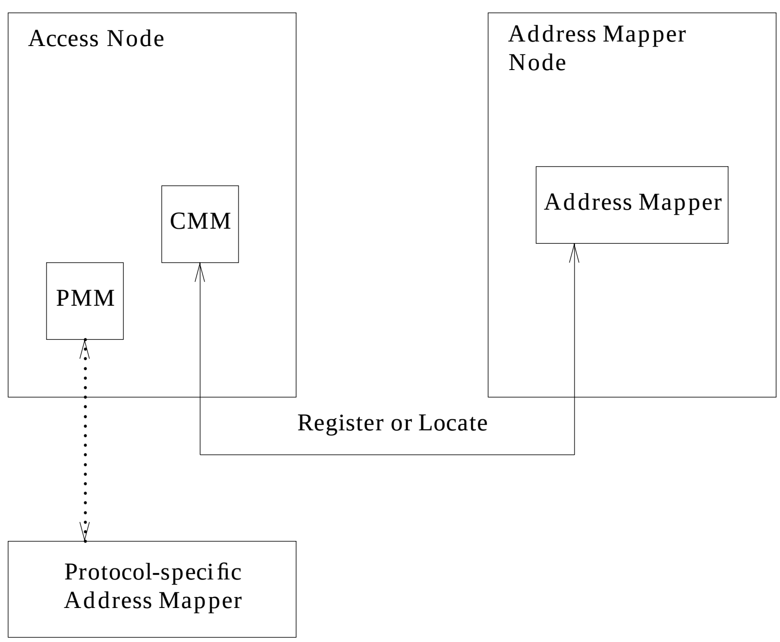

Figure 5-5 Address Mapping Protocols

Chapter 6

MPTN Benefits and Limitations

This chapter discusses the benefits that an MPTN based solution might offer to a customer. It also discusses the limitations of MPTN.

6.1 Benefits of MPTN

By defining the TLPB and a common set of compensations to make up for differences in the various transport providers, MPTN breaks the binding between an application and the underlying communications protocol. This enables the choice of applications and the choice of a communications protocol to be made independently of each other, resulting in several benefits:

- For the end user:
  - — Existing applications can be run over additional network types, expanding the scope of existing applications and giving the end user a wider choice of possible applications.
  - — An application can be chosen based on its merits, since the choice is no longer restricted to the set of applications that can run over the installed communication protocol(s).
- For the application provider:
  - — Application writers developing applications that use communication services can select the API they use based on the functionality the API provides, not based on the API the installed communications protocol(s) support.
  - — Existing applications can be run over additional network types, expanding the market for those applications. Therefore, application providers can concentrate on improving their product and providing additional functions rather than in developing different versions of their product to run on different communications protocols.
- For the network administrator:
  - — Selection of a communication protocol can be based on the merits of that protocol, not on the number of applications available for that protocol.
  - — Networks can be consolidated, and the number of communication protocols to be managed can be reduced while still supporting all existing user applications.
  - — A network can be changed without affecting existing applications. If a newer, better communications protocol comes along, the old protocol can be replaced, and all existing applications will continue to work.

6.2 Limitations of MPTN

The general limitations of MPTN are those listed in Appendix A. What follows are some technical limitations which some applications may expect to encounter.

MPTN is intended to allow end-user applications to run over addition network types. If the application is a network configuration tool, or an administration tool, or somehow relies on the transport functions of a specific transport provider type, the application or tool may not run using MPTN. Some examples will help to clarify this:

- For an AF-INET Sockets transport user, the TCP state can not be be monitored. Since the connection does not use TCP, there is no TCP state. Network administration tools, such as netstat, which show the state of the TCP connection, have no state to show. Implementations may show some default state, such as CLOSED.
- In general IP options are not supported for AF-INET transport users. These options include loose source routing, strict source routing, record route, and timestamp. These options only make sense to IP routers, and since there is no underlying IP, there are no IP routers.
- For NetBEUI transport users, SEND timeouts and related features are not supported. There is no underlying NetBIOS protocol to perform these functions.

MPTN does not attempt to solve the problem of unlike applications communicating, for example, an SNA application communicating with a AF-INET Socket application, or a NetBEUI application communicating with an IPX/SPX application. This problem is outside the scope of the MPTN architecture.

Appendix A

Frequently Asked Questions

The following concerns have been raised in the process of developing XMPTN, with responses from those who support the MPTN approach.

This appendix is included as background information, for the benefit of those interested in reviewing the MPTN approach in the light of these concerns and of alternative solutions.

1. **Concern:**

   MPTN is too complex a solution.

   Response:

   The protocols involved in MPTN are not complex. Experience shows that it is possible to build prototypes of specific protocol pairs in a matter of months, including associated native-to-nonnative gateways.

   What is complex is the extent of the underlying problem: the number of different transport user and transport provider types, and the fact that it is easy to get the two confused when talking about solutions. For example, people find it easy to get socket-over-SNA and APPC-over-TCP prototypes confused because one is the opposite of the other.

   Given the number of different transport user and transport provider types, solving each case in an ad hoc manner (for example, more RFCs like RFC 1006) would yield a considerably more complex overall solution than MPTN, which takes advantage of the commonalities that exist between the different problems. If the PMM component associated with a particular transport provider is written in terms of the compensations it needs to provide for particular transport user functions, rather than in terms of what it needs to do for a particular transport user type, then new transport users can be run over the transport provider with little or no change.

2. **Concern:**

   There already exist ways other than MPTN for performing compensations, like the IETF RFCs 1001 / 1002 / 1006.

   Response:

   This statement is true for *some* protocol combinations, particularly those with a TCP / IP transport provider. Understandably, there appears to be no desire to replace such means, where they have already been invented and widely implemented, with a general approach. However, all combinations have not been covered by RFCs.

   The RFCs are specific for every transport user to TCP / IP transport provider pair (RFC1006 for OSI applications, RFC 1001 / 1002 for NetBEUI applications). MPTN can be used for many more permutations of the transport-user / transport-provider.

   RFC 1006 does not describe a mechanism for interconnecting OSI applications running over TCP with matching OSI applications running on a native OSI stack. MPTN does include mechanisms for constructing native-to-nonnative gateways for this purpose.

3. **Concern:**

   MPTN leads away from standards, specifically OSI.

   Response:

   See Section 2.4 on page 9. Vendors cannot control the way users move to OSI. It can be argued that MPTN makes the migration easier by allowing them to move to OSI AND keep their existing investment in applications. Also, MPTN does make it possible for vendors to take a two-step migration: first by replacing proprietary networks with TCP/IP, keeping the associated applications; then substituting TCP/IP for OSI by replacing the transport provider.

4. **Concern:**

   MPTN preserves proprietary interfaces; it does not attract people to move to open systems.

   Response:

   Again, it is questionable whether X/Open's answer to the user requirement for coexistence with existing networks should be "move to XTI and OSI". The users should be free to make their decision themselves. Generally, their investment in other interfaces and protocols is too large to be replaced with exclusively open systems in one move. MPTN would actually make it possible to reduce the number of different transport providers an enterprise has to support, thus making the eventual transition to OSI easier.

5. **Concern:**

   MPTN is ten years too late. By now, all the vendors in the business have solved the interoperability problems for their users through development of well-functioning gateways. There is no need to standardise such gateways.

   Response:

   The user community, as demonstrated by the SOS letter and the X/Open *Xtra 1992* requirements process, does not appear to think that the problem has been completely solved.

   Perhaps the users feel that these gateways are proprietary to particular vendors, and thus do not preserve their ability to make open choices.

6. **Concern:**

   X/Open should not be specifying protocols; such work is for formal standards organisations to undertake.

   Response:

   In the past, X/Open has both specified (meaning published) and developed protocols. The (PC)NFS, SMB and XNFS specifications all include definitions of non-OSI protocols. For the SMB specification, a new security-related protocol was developed by the PCIG. Besides, the fact that X/Open has not done something so far should not be an automatic inhibitor for us doing it now, if the users need it. Furthermore, no body other than X/Open is likely to take up MPTN, since MPTN is a general solution addressing interoperability between various transport protocols, while all the other bodies care about one or at most two protocol suites at a time. Publishing MPTN specifications would certainly spotlight X/Open as the leading interoperability enabler in the industry.

7. **Concern:**

   The proposals for MPTN propose modifications to XTI; this is unacceptable if such modifications result in incompatibility of XTI with its existing applications.

   Response:

   All syntactic changes proposed for XTI have been withdrawn, except possibly the definition of a few XTI-level options. These could be made as part of the XTI revision in the later part of 1993.

   For MPTN, a desired change to XTI is for a semantic change for the meaning of *t_open*( ), so that the *name* provided by the transport user can be taken as a specification of the transport characteristics it requires, even if those characteristics are not directly supported by the available transport provider(s). there is at least one alternative way to handle the selection of transport provider. The alternatives need to be evaluated. Those who currently prefer a semantic change for the meaning of *t_open*( ) believe this would be the least disruptive to existing programs, providing them with access to a broader class of transport networks without programming changes.

8. **Concern:**

   The cost of this greater connectivity to users who are not using MPTN is larger kernels, eventually leading to slower performance. Since it seems that MPTN protocol data will have to be carried in what usually would be the user data stream, above the XTI transport provider, this is very expensive data through the individual stacks on the target, the network, and the routers.

   Response:

   It is not possible to add new function without increasing code size. Whether the additional function is worth it is a judgement both vendors and users will make.

   The particular concern here appears to be the users that are not using MPTN. When MPTN is not needed (transport users match transport providers) there is no overhead in terms of line flows, protocol, or function. The native stack can be used directly. Whether the existence of MPTN in the box will affect native transport users is an implementation issue. There appears to be no experience of this so far.

9. **Concern:**

   The MPTN architecture document does not mention Quality of Service - nor an OSI "service metric".

   Response:

   This was not included in early drafts of this document, based on an assumption that this information would be too detailed for the level it is aimed at. Quality of Service is now mentioned in this Guide.

10. **Concern:**

    There are significant concerns over the MPTN gateway performance and cost. Path lengths increase with MPTN Compensation, and the gateway is responsible for connection establishment, determining the compensation required, and needs address mapping. The belief is that this makes connection set up phases very complex and time consuming, with poor processing performance.

    Response:

    Performance considerations are relative. Clearly, performance when using a transport gateway will not be as good in processing terms as when running directly on the same network. However, transport gateways will perform better than application gateways, which require much more complex transformations and state machines.

    MPTN gives a network designer choices: run a distributed application on the same network (which may mean installing additional networking software in some endpoint machines), or install MPTN gateways between networks, allowing the endpoint machines to run without additional protocols. Clearly, the first will perform better than the second, since no processing is required in the middle. But there are users for whom the gateway is believed to be the better choice - for example, for those who need only occasional communication, or those for whom a single end-to-end protocol is not feasible because networking choices are under the control of different organisations.

    All processing costs something. Given that most compensations involve inserting a one-byte label in front of the user data, it does not seem reasonable to regard this as prohibitive.

    The normal send/receive processing has been kept as simple as possible, perhaps at the expense of doing more during connection establishment. However, the connection setup phase is not believed to be "very complex." The connection establishment process in the gateway is the same as that in the access nodes, with the possible addition of accessing the gateway routing tables in intermediate gateways. Also, for the MPTN connect message, the gateway may need to modify only one field before passing it on to the destination or next gateway. Address mapping only occurs in the gateway on the edge of the destination network, not in every gateway along the path in the multiple gateway case.

11. **Concern:**

    With format and protocol conversion, every packet will have to go through a transformation (in the MPTN gateway), and each packet potentially has an effect on the state of each connection (requiring state information to be maintained). The performance problem persists on established connections.

    Response:

    The transformation consists of removing the MPTN label, if the next network supports the function associated with the packet natively. Otherwise, the gateway just forwards the packet. The underlying transport providers do need to maintain the usual state for connection endpoints. The additional MPTN state required is minimal.

12. **Concern:**

    The gateway, as a transport relay, has to acknowledge and relay data as well as handle flow control. If state information and data buffers have to be maintained for every connection, this makes the gateway costly in memory requirements.

    Response:

    It is true that the gateway serves as the transport endpoint for the individual transport connections that it joins. This allows the gateway that is slowed down by flow control on one side to apply back pressure to the other side, causing end-to-end flow control without new flow control protocols.

13. **Concern:**

    With transport relays, additional transit delays occur with recombination of TPDUs. This makes them inappropriate for high throughput links (which are today's trend).

    Response:

    MPTN does not reassemble segments in the gateways. The reassembly takes place in the destination. Therefore there are no additional transit delays associated with recombination of TPDUs.

14. **Concern:**

    Gateway routing protocols rely on ISO IRDP (DIS 10747). This protocol is still unstable, with known defects.

    Response:

    The following is taken from a report by the IRDP Editor (Charlie Kunziger) on the Routing Meetings within ISO SC6 / WG2, London, UK:

    > Although the formal DIS ballot does not close until April 22, 1993, the US and UK had completed their ballot comments and were prepared to discuss them at the London meeting. US and UK comments will accompany "NO" votes; all other national bodies anticipated that they would return "YES" votes.
    >
    > The national bodies in attendance agreed that it was appropriate to discuss the US and UK comments at the meeting, with a view towards resolving them, thus expediting the publication of the final text of the International Standard.
    >
    > All outstanding major comments upon which the US and UK "NO" votes were based were resolved to the satisfaction of all national bodies: that is, the proposed resolutions of comments would not cause any any other national body to vote "NO".
    >
    > A poll of the delegates indicated that no additional "NO" votes were anticipated. Hence, it was noted in the plenary session that if there are no other "NO" votes at the formal close of ballot, the ISO directives permit the editor to amend the DIS text in accordance with the agreed-upon resolutions, and to submit the revised text for publication as an International Standard. That is, the Working Group agreed that all technical discussion on IDRP has been completed, and the final text is expected to be submitted for publication as an International Standard in the June timeframe.

    The ISO IDRP protocols achieved International Standard status in October 1993.

15. **Concern:**

    How well will the MPTN gateway react in high-availability environments?

    Response:

    The full MPTN gateway architecture includes parallel gateways so that a single point of failure can be avoided.

16. **Concern:**

    MPTN gateways do not provide end-to-end reliability, in that data received on the incoming reliable connection could be lost before it is sent on the outgoing reliable connection. When data is lost or corrupted, it will not be flagged by the gateway. The MPTN access node may have the same problem.

    Response:

    The current proposal to resolve this concern is to add a new compensation for End-to-End reliability. Data can only be lost/corrupted as it passes through the MPTN Gateway relay, because the transport provider connections must be reliable (by definition). So, this compensation is only applied if requested by the transport user and the connection passes through an MPTN Gateway.

    This new End-to-End reliability compensation includes an MPTN sequence number and MPTN checksum. The MPTN sequence number and checksum are added at the first MPTN node encountered, and removed/verified at the last MPTN node.

17. **Concern:**

    Can the transport user and transport provider pieces of the protocol stack be provided by different vendors?

    Response:

    The ability to mix user and provider parts from different vendors was not a goal in the design of MPTN, and is not part of the MPTN Architecture.

    The MPTN specifications define a model, including the CMM, TLPB, BSPB, and PMM. This is only a model; an implementation is not required to follow the model. In addition, the TLPB and BSPB are protocol boundaries, not interfaces. If an MPTN implementer follows the model, they may use whatever transport interface they choose based on several factors, including ease of implementation, performance, and marketability.

18. **Concern:**

    Will XMPTN implementations from different vendors interoperate? For example, will an implementation of SNA over TCP/IP from vendor A interoperate with an implementation of SNA over TCP/IP from vendor B?

    Response:

    Yes, they will.

Glossary

abortive termination/release
An abrupt termination of a transport connection, which may result in the loss of data.

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 be legal in another protocol, MTPN qualifies all transport addresses with an address type.

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

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

connection data
User Data that is sent as a part of the transport connection request.

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

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

duplex termination
A method of terminating a connection where both directions of a full-duplex connection are closed at the same time.

expedited data
Data that is considered urgent. Such data may be delivered ahead of but no later than all normal data that preceded it, and is not blocked by normal flow control mechanisms. The specific semantics of expedited data are defined by the transport user, matching what could be expected from well known transport providers.

Internet, the
The cooperative virtual network that uses the TCP/IP protocol and includes the ARPANET, MILNET and NSFnet. It provides universal connectivity and reaches many universities, government, military and commercial establishments.

Internet Protocol Suite
(IPS) The protocol suite used by ARPANET that is composed of the Transmission Control Protocol (TCP) and the Internet Protocol (IP). It is often referred to as TCP/IP.

MPTN
Multiprotocol Transport Networking

MPTN Access Node
A node which has MPTN components installed.

MPTN Address Mapper
An MPTN component that provides address resolution for non-native transport addresses.

MPTN datagram
A datagram that carries an MPTN header as part of the data.

multicast
A transport mechanism in which a message originated from a single source can be delivered to multiple, specified destinations simultaneously.

native
A transport user that has both an address type and transport characteristics of the transport provider serving it. No MPTN protocols are used for data transfer.

native node
A node that does not contain MPTN function.

native transport address
A transport user address, the address type of which corresponds to the transport protocol on which it is being used; for example, an SNA name that is being used within an SNA network.

non-native network
A network whose addressing structure is different from the transport user's addressing structure.

non-native transport address
A transport user address whose address type is different from the transport provider address type; for example, an OSI address that is the target of a connection request using SNA transport.

orderly termination/release
A procedure for gracefully terminating a transport connection with no loss of data.

record data format
The record boundaries that must be maintained between the source and destination end points for the data being transmitted.

record delivery
A form of data delivery in which data being exchanged has record boundaries that must be maintained.

service mode
Service mode is used by transport users to request characteristics that must be maintained on a given connection or datagram transmission. Each network protocol has its own way of requesting these characteristics which must be mapped to the MPTN service mode.

session outage notification
Notification given to the transport user that the transport provider connection has broken. Some transport providers require that this notification be given immediately, or within a very short time period, of the transport provider connection breaking.

simplex termination
A method of terminating a connection where only one direction of a full-duplex connection is closed at a time (the send direction). When the user issues a close, only the send pipe is closed, the receive pipe remains up. This pipe is closed when the communication partner

closes its send pipe.

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

stream delivery
A form of data delivery in which data being exchanged is simply a string of bits (or stream), there are no record boundaries to preserve.

termination data
User Data that is sent as a part of the transport termination request.

transport
The services and protocols associated with end-to-end use of a network, for example, OSI layer 4, TCP and UDP.

transport network
A network accessed via the transport services provided by a specific transport provider

transport provider
The set of protocol and MPTN functions that provide a defined set of transport services to transport users. Each transport provider is based on a specific transport protocol.

transport user
An application program, service, or higher layer protocol that uses transport services to convey data through a network.

transport user address
A transport address that allows a transport user to be located in the MPTN network but is not necessarily used by the underlying transport provider. A transport user address is a transport address and not a higher-level application directory name.

Index