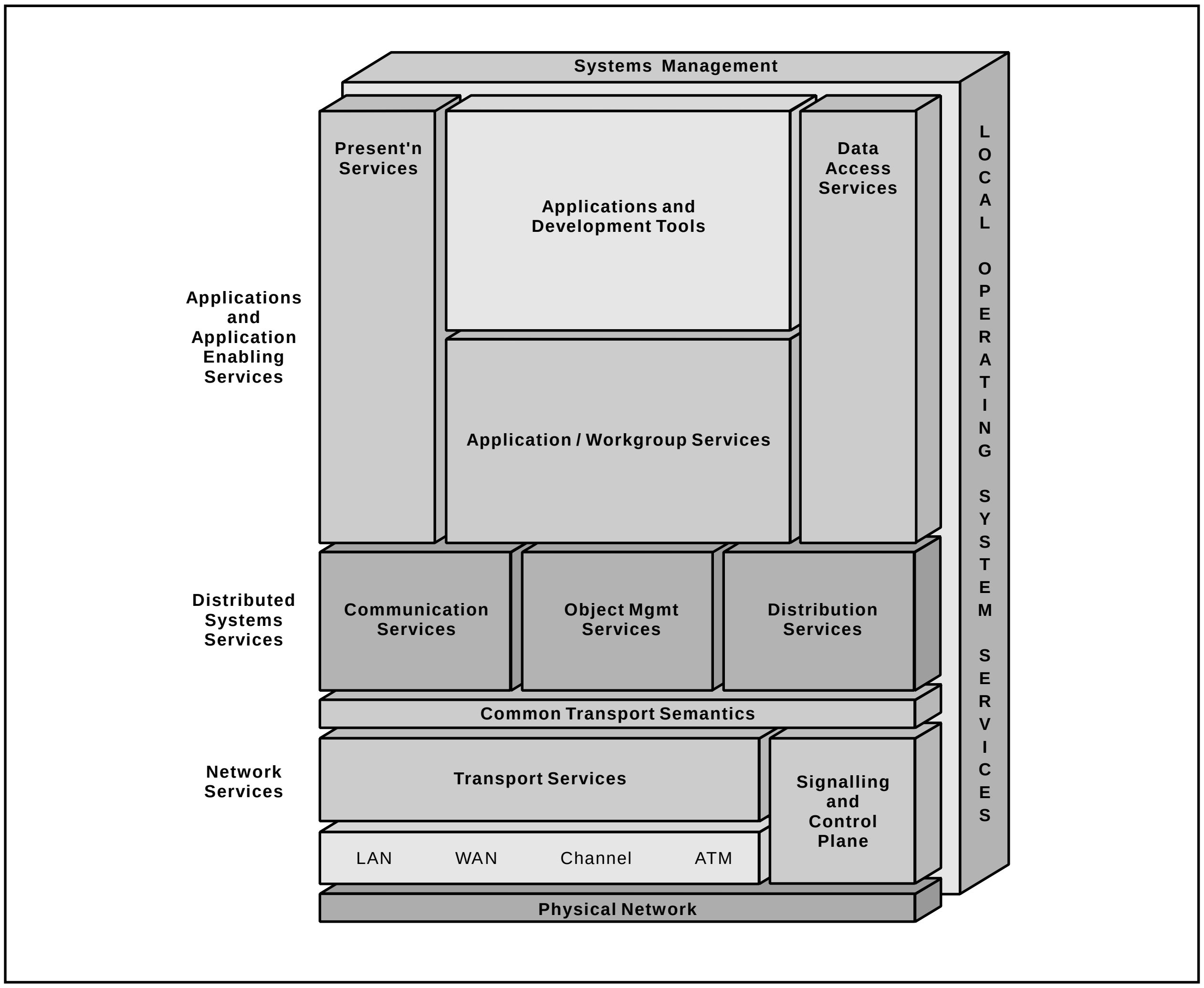
Open Blueprint

Network Services



Systems Management
Present'n Services
Applications and Development Tools
Data Access Services
Applications and Application Enabling Services
Application / Workgroup Services
LOCAL OPERATING SYSTEM SERVICES
Distributed Systems Services
Communication Services
Object Mgmt Services
Distribution Services
Common Transport Semantics
Network Services
Transport Services
Signalling and Control Plane
LAN
WAN
Channel
ATM
Physical Network

GC23-3932-01

Open Blueprint

Network Services

GC23-3932-01

About This Paper

Open, distributed computing of all forms, including client/server and network computing, is the model that is driving the rapid evolution of information technology today. The Open Blueprint structure is IBM's industry-leading architectural framework for distributed computing in a multivendor, heterogeneous environment. This paper describes the Network Services component of the Open Blueprint and its relationships with other Open Blueprint components.

The Open Blueprint structure continues to accommodate advances in technology and incorporate emerging standards and protocols as information technology needs and capabilities evolve. For example, the structure now incorporates digital library, object-oriented and mobile technologies, and support for internet-enabled applications. Thus, this document is a snapshot at a particular point in time. The Open Blueprint structure will continue to evolve as new technologies emerge.

This paper is one in a series of papers available in the *Open Blueprint Technical Reference Library* collection, SBOF-8702 (hardcopy) or SK2T-2478 (CD-ROM). The intent of this technical library is to provide detailed information about each Open Blueprint component. The authors of these papers are the developers and designers directly responsible for the components, so you might observe differences in style, scope, and format between this paper and others.

Readers who are less familiar with a particular component can refer to the referenced materials to gain basic background knowledge not included in the papers. For a general technical overview of the Open Blueprint, see the *Open Blueprint Technical Overview*, GC23-3808.

Who Should Read This Paper

This paper is intended for audiences requiring technical detail about the Network Services in the Open Blueprint. These include:

- Customers who are planning technology or architecture investments
- Software vendors who are developing products to interoperate with other products that support the Open Blueprint
- Consultants and service providers who offer integration services to customers

Contents

Figures

Summary of Changes

The changes are for this revision are described below:

- This publication was previously named *Open Blueprint Transport Services*. The title was changed to better represent the publication's scope.
- The description of TCP/IP Transport Services support has been expanded to include:
  - Dynamic TCP/IP support (Dynamic Host Configuration Protocol (DHCP) and Dynamic Domain Name System (DDNS))
  - Mobile IP
  - Secure IP standards (Secure Sockets Layer (SSL) and Secure HyperText Transfer Protocol (SHTTP))
  - Real-time support protocol information (Resource Reservation Setup Protocol (RSVP) and Real-Time Protocol (RTP))
  - Next Generation IP (IPv6)
- References to OSI have been removed

Open Blueprint Network Services

Network Services provides a common view of transport semantics to make higher-level distributed systems services and application enabling services independent of the underlying transport network. Network Services are:

- Common Transport Semantics

  Common Transport Semantics supports protocol-independent communication in distributed networks.

- Transport Services

  Transport Services supports the protocols needed to transport information from one system to another. These protocols include TCP/IP, SNA/APPN, NETBIOS, and IPX.

- Signalling and Control Plane

  The Signalling and Control Plane provides the ability to establish subnetwork-specific connections.

- Subnetworking

  Subnetworking provides a structure to let networks evolve to and exploit new high-speed, highly-reliable transmission technologies without sacrificing business application and network investments.

This paper provides technical details about Common Transport Semantics and Transport Services. For information about the Signalling and Control Plane and Subnetworking, see the *Open Blueprint Technical Overview*.

Common Transport Semantics (CTS)

Common Transport Semantics (CTS) insulates the higher-level services of the Open Blueprint from the underlying transport network by providing a common view of transport protocols. This common view, coupled with a standard set of compensation mechanisms, enables all higher-level services to be transport-independent, so that different transport network drivers can be plugged in under a common implementation of those services.

CTS enables a network to be based on the one protocol (or few protocols) best suited to the enterprise network requirements. Then, by providing a common view of networking protocols, this structure makes application environments independent of the underlying network. For example, data needing conversational, remote procedure call, or messaging and queuing services can be carried over any network protocol. New applications can be added to any network without requiring installation and maintenance of additional networking protocols. One need not build parallel networks for different applications; all forms of management (problem determination and configuration) are simplified and less costly because fewer transport types are used. If you are unable to converge parallel networks into a single one, CTS allows you to interconnect the different network types using standard applications.

Support for multiple protocols can be achieved in several ways; the Open Blueprint provides the context for discussing them. The CTS layer in the Open Blueprint provides support for multiple protocols at the transport layer. CTS supports all of the basic transport functions of the underlying transport providers in the Open Blueprint. If needed functions are missing from any of the transport providers, CTS itself provides those functions. CTS function can be implemented in different ways depending upon the situation:

- CTS supports an installed application program that is native to a specific transport protocol. In this case, CTS is a conceptual structure that adds no line flows or processing overhead.

- CTS function can be implemented using industry-standard compensation methods for specific protocol combinations, such as the Internet Engineering Task Force (IETF) Requests for Comment (RFC) 1646 and 1647 for TN3270E gateway, and RFCs 1001 and 1002 for NetBIOS over TCP/IP. The TN3270E gateway allows TCP/IP Telnet clients to communicate with SNA hosts. NetBIOS over TCP/IP allows NetBIOS programs to run over a TCP/IP local area or wide area network.
- The Multiprotocol Transport Networking (MPTN) architecture formats and protocols deliver CTS function in situations where the installed application programs are not native to the installed transport protocol, or where multiple transport protocols are installed. Compensation for missing functions is available in those situations. For example, the MPTN architecture defines how SNA can be the transport provider for sockets applications and how TCP/IP can be the transport provider for Common Programming Interface for Communications (CPI-C) applications. In 1995, X/Open adopted IBM's contribution of MPTN architecture as an industry standard for mixed-protocol internetworking.

The MPTN architecture, as implemented by the IBM AnyNet family of products, provides a significant aspect of the common transport semantics (CTS) layer of the Open Blueprint. MPTN provides the functions to allow transport users to access non-native transport providers. For a transport API such as sockets or NetBIOS, the transport user is the API processing layer. For an API such as CPI-C, which exposes higher-level SNA functions, the transport user is the part of the SNA stack above the transport layer. The communication APIs above the transport users are shown in Figure 1. Details of the MPTN functions are described in the next section.

Figure 1 shows some examples of transport users. The rows of triangles at the top represent higher-level APIs, and the rows of arrows toward the bottom represent a semantic interface to MPTN functions. There is no standard interface to the MPTN functions. Implementations of MPTN use system-specific interfaces to access each of the transport stacks. The different sizes of the boxes represent differences in the amount of function in the respective system libraries. Thus, there are more functions in the system library to support an interface such as CPI-C with upper-level services than there are to support a transport-level interface such as sockets.

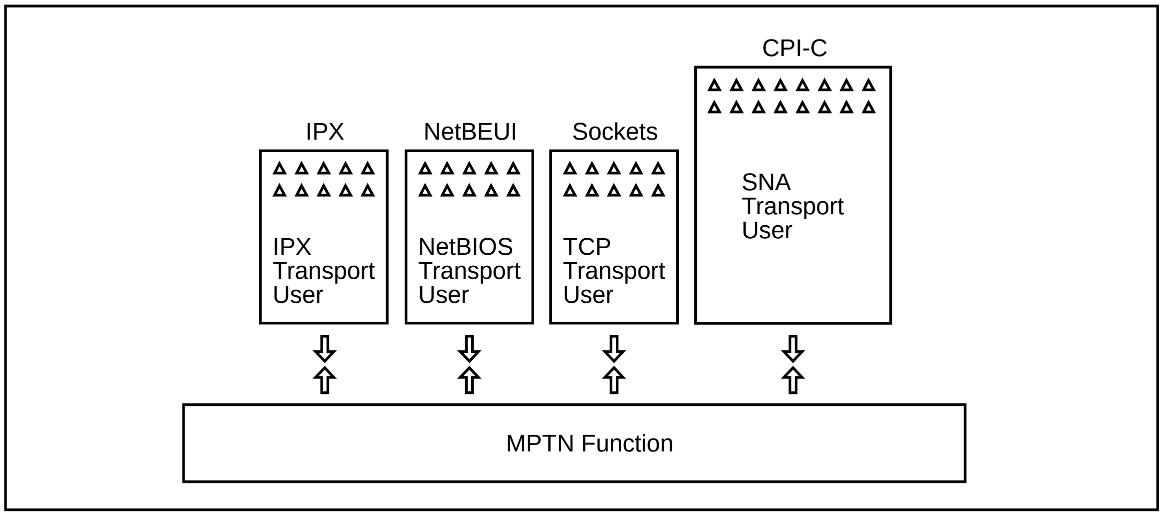


Figure 1. Examples of MPTN Transport Users

CTS is also used by upper layer applications and services to access the Signalling and Control Plane to set up connections across switched networks such as ATM or plain old telephone service (POTS), and to control low-level multiplexing, such as that employed in H.320 video conferencing. For example, the Telephony and Collaboration resource managers use the functions of the Signalling and Control Plane. The Telephony resource manager uses it to set up telephone calls, and to manipulate calls that are in progress. The Collaboration resource manager uses the Signalling and Control Plane to set up switched connections among multiple participants in a collaborative session and to control the low-level multiplexing of voice, video, and data, as in the case of H.221 multiplexing (part of the H.320 video- conferencing framework).

Common Transport Semantics allows mixing and matching of application requests to lower layer networking services by adding compensations where needed, for example, to run sockets applications on an SNA network. If the application request already matches the lower layer service, no compensation is needed and CTS is passive. Today, no mismatches have been identified between application requests and Signalling and Control Plane services, so CTS plays a passive role in all interactions with the Signalling and Control Plane.

Multiprotocol Transport Networking (MPTN)

Multiprotocol Transport Networking (MPTN) architecture supports:

- **Mixed Protocol Networking**. Running application programs that were designed to operate over one transport protocol (such as SNA, NetBIOS, IPX, or TCP/IP) over additional transport protocols
- **Network Interconnection**. Connecting matching application program partners across different types of transport networks

MPTN Architecture Terminology

As illustrated in Figure 2, there are two sets of relationships that are important to this architecture: peer relationships and relationships between transport users and transport providers. The term *matching* describes a horizontal relationship between peers. In Open Blueprint terms, common communication protocols are required for interconnection. In the MPTN architecture, transport users must always have matching protocols. For example, two sockets programs match or two CPI-C programs match, but a sockets program and a CPI-C program do not.

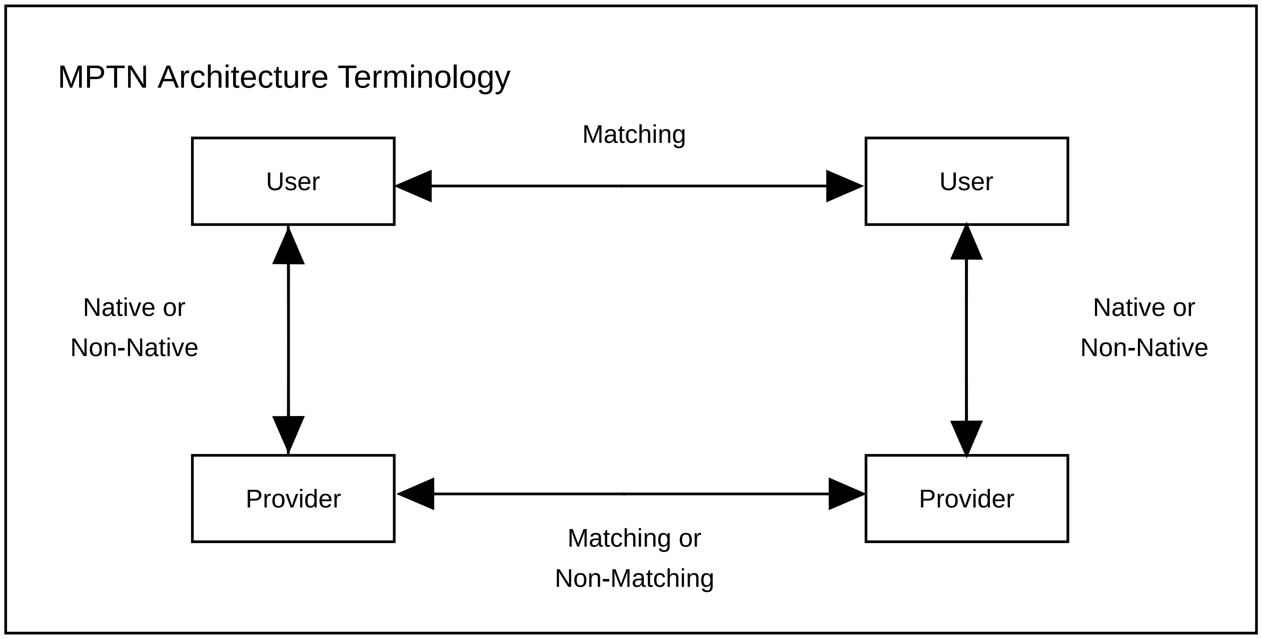


Figure 2. Matching and Native Protocols

Requirements for Applications over Non-native Transports: Businesses want the best solutions for their business problems. They do not want to be forced into transport protocol choices based solely on their application program choices, nor do they want to be prevented from acquiring the best available application program because it requires a transport protocol they do not support. Thus, businesses have requirements for running application programs without change over one or more transport types for which the programs were not originally developed.

Requirements for mixed-protocol networking include:

- Selection of applications and resource managers can be separate from the selection of transport protocol. Often, different organizations own these two responsibilities, and they need the ability to optimize their decisions independently.
- Selected applications and resource managers must run on either a native or non-native transport protocol stack with the same code, because they are often purchased separately to run on an existing installed base of different transport protocol stacks.
- When multiple transport protocols are installed on the same machine, transport protocol selection must be transparent to the application program. Otherwise, multiple copies of the same application program must be installed, and the resulting complexities must be managed.

Requirements for Network Concatenation: In today's business environment, organizations need to exchange information with other organizations such as vendors, suppliers, other departments, or subsidiaries without installing additional transport protocols. Therefore, they need the ability to interconnect nodes between which no complete single protocol path exists.

For example, a corporation might have a set of independent departmental networks that run different transport protocols. Eventually, the separate networks need to be interconnected for occasional interactions. With network concatenation, two different networks can be interconnected by installing new, non-native application programs on one side and an intervening transport gateway between the two networks. The other network can run without change. When the frequency of data exchange between those two networks increases, the corporation has the option of eliminating the transport gateway by providing a single transport network, without having to change the application programs.

Thus, there is a requirement to accommodate two different approaches to interconnecting networks:

1. Run all upper-level services over a single transport protocol.

   As an example, consider two different approaches:

   - Running World Wide Web HyperText Transfer Protocol (HTTP) or other TCP higher-level services, such as File Transfer Protocol (FTP) or Telnet over SNA.
   - Replacing other networking protocols with TCP/IP and continuing to run existing application programs and new TCP application programs over the Internet or a private intranet.

   In both cases, organizations require the ability to protect their current investment in networks and applications.

2. Permit different transport protocols in different portions of the composite network, while maintaining availability of common upper-level services.

   This requirement includes allowing an application program that runs over one transport network to interoperate through a gateway with a matching application program running on another transport stack. For example, a Distributed Computing Environment (DCE) application program running over TCP/IP should be able to interoperate with a DCE program running over SNA.

The MPTN Architecture As the Solution: The requirements identified in the previous sections can be condensed into the following requirements which the MPTN architecture satisfies:

- Network users can add a new application program based solely on how well it meets their business needs without requiring their network administrator to install a specific transport network. Off-the-shelf application programs can now be used in new environments.
- Network planners can choose the transport network best suited to their business needs without constraining their users' choice of application programs. Non-native application programs can be added to existing networks.

  Furthermore, network planners have the flexibility to configure their networks so that different transport networks can be deployed as needed, while allowing users and matching application programs on different transport networks to communicate.
- Network managers can change from one transport network to another while maintaining their investment in existing application programs. They can reduce the number of installed transport protocols, thus simplifying maintenance and management of the network.
- Application designers are free to develop new application programs on an application programming interface (API) based solely on the function of that API, and they are not limited to an API that accompanies the installed transport network.
- Network managers can manage their overall network using the management tools native to the various transport networks.

The MPTN architecture addresses these customer requirements with MPTN *access node* and MPTN *transport gateway* components. MPTN access nodes provide the platform for running application programs over non-native protocols, and MPTN transport gateways interconnect unlike transport networks.

As shown in Figure 3 on page 8, MPTN architecture encompasses five basic networking configurations.

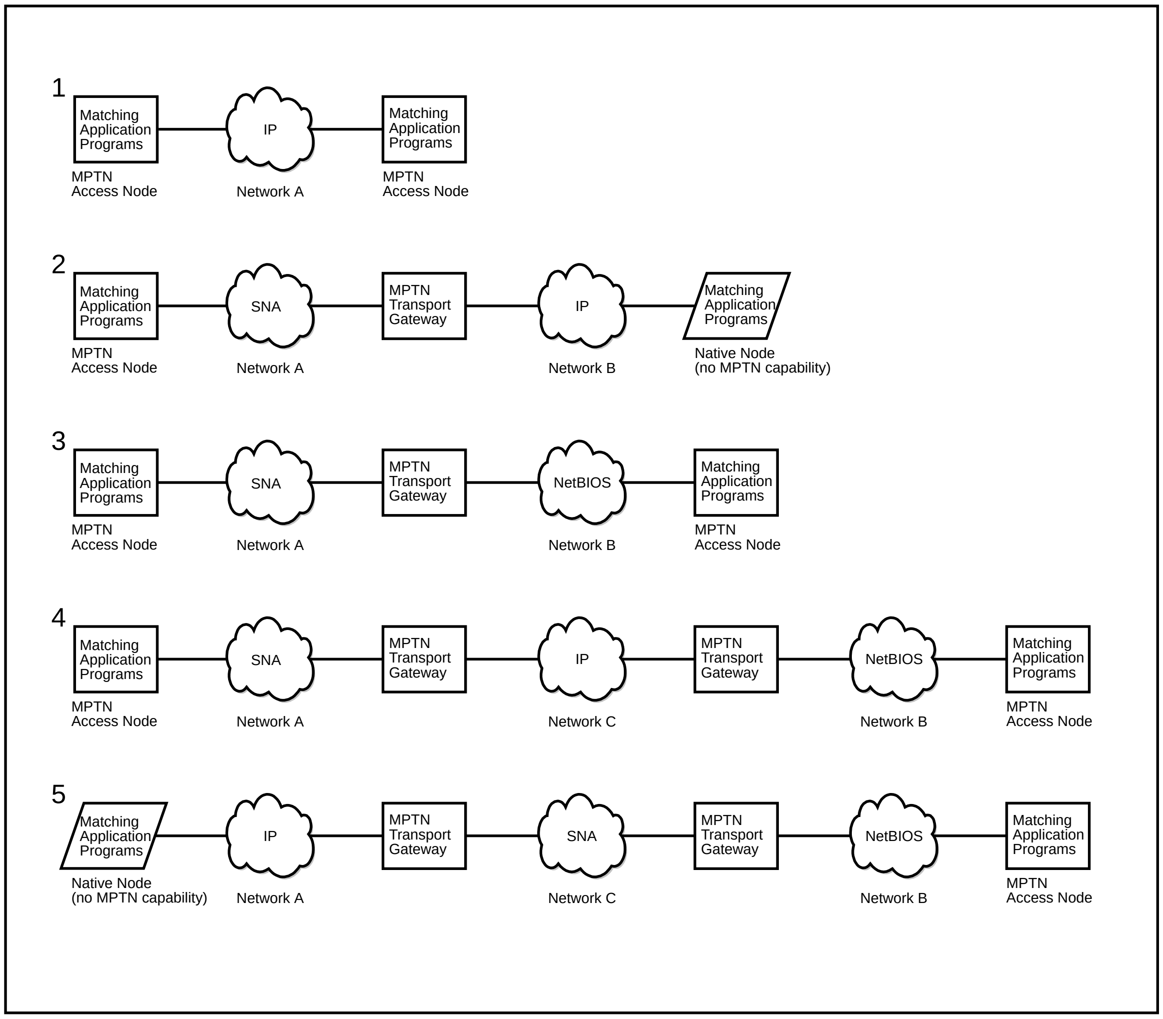


Figure 3. Five Basic Networking Configurations

- **Configuration 1**. Illustrates that both MPTN access nodes running matching application programs are connected without an MPTN transport gateway.
- **Configuration 2**. Illustrates an MPTN access node connecting through an MPTN transport gateway to a native node that is running a matching application program. For example, the MPTN access node could be running a Web server over SNA, and the native node could be a UNIX workstation running a Web browser.
- **Configuration 3**. Illustrates both MPTN access nodes running a non-native application, but they are using different transport networks. This could be a case where a sockets-based resource manager such as remote procedure call (RPC) is running over SNA in network A and over NetBIOS in network B.
- **Configuration 4**. illustrates the fact that the MPTN architecture does not restrict the number of networks that can be traversed, and it even provides protocols between the MPTN transport gateways to dynamically learn about the interconnected networks. This configuration could support the same access nodes as in configuration 3 with the Internet, for example, interposed between the SNA network and the NetBIOS network.
- **Configuration 5**. Illustrates that both end nodes using different transport networks are connected through two MPTN transport gateways.

MPTN Transport Services

By making a general set of transport services available through CTS, the MPTN architecture enables higher-level resource managers, such as Remote Procedure Call (RPC), to work over multiple transport protocol stacks, even though it was originally written to work over TCP/IP only.

In theory, two approaches exist for deciding what specific functions to provide in CTS:

- **Select only the functions that are common to all of the transport technologies being considered**. Such an interface would have minimal function and therefore would require the user to add functions that are already provided in many protocols.
- **Select all generally useful and necessary functions and provide general compensations for missing functions**. This approach was chosen to satisfy the above requirements concerning RPC. The number of compensations is relatively small.

The MPTN architecture does not include functions that provide transport-protocol-specific information (such as adapter status) to the transport user.

Connection-Oriented Services: Connection-oriented services establish a logical full-duplex connection between two partners for the duration of the period they communicate with each other. The connection provides reliable, in order (first-in, first-out) delivery of data.

There are three distinct phases in a connection-oriented data exchange:

- Connection-Establishment

  The logical connection between two partners is established during the connection-establishment phase. The transport users of some communication protocols send connection data (for example, a BIND in SNA) to specify the characteristics of the connection; for other communication protocols (for example, TCP/IP), no connection data is sent. The MPTN architecture supports connection establishment with or without connection data; the user can accept or reject the connection after examining the connection data.

  The MPTN architecture supports a general service mode that can map to SNA's class of service or TCP/IP's type of service, to allow a user to control the quality of service over one or more non-native transport providers.

- Data Transfer

  Reliable data transfer is a characteristic of the data transfer phase that is provided by the transport provider underlying MPTN. Other characteristics include whether a stream or record model is used and whether expedited data is supported.

  Some user applications and higher-layer protocols assume data record boundaries are preserved by the underlying transport provider. Others send data as a byte stream with no record boundary delineation. Both data-delivery models are supported by the MPTN architecture. For record models, the maximum record sizes can vary among the various transport providers. The MPTN architecture allows the transport user to select the maximum record size based on what it expects from its native transport provider.

  Some user applications and other higher-layer components send expedited data. The MPTN architecture assures that expedited data:

  – Is not subject to the same data flow control as normal data

  – Is not delivered later than any normal data sent after it, but can be delivered ahead of normal data sent before it

  – Can be delivered to the transport user even if it is not receiving normal data

  - Can be distinguished from normal data when sent or received.

- Connection Termination

  During connection termination, some transport users expect to be able to deliver data related to the termination process (such as the reason for the termination) to the partner. Other transport users do not require this ability. Some transport users expect to terminate one direction of the connection at a time (simplex termination), while others expect to terminate both directions of the connection at once (duplex termination). Similarly, some transport users expect the connection termination data to stay in-line behind data sent earlier (orderly termination), while others permit it to overtake data sent earlier (abortive termination). The orderly termination of the connection, which assures the delivery of all previous sent data, might be preferable. All of these connection-termination semantics are provided by the MPTN architecture.

Connectionless Services: Connectionless service is the mode of data transfer in which different units of data are passed independently through the network. Delivery is provided on a best-effort basis, so undetected packet loss can occur, for example, if the network becomes congested. There is no guarantee that data units will be delivered in the same sequence that the sender issued them or that duplicates will not be delivered. The basic unit of connectionless data transfer is the *datagram*, which is a data packet that carries information sufficient for routing from a source to its intended destination.

Different transport providers have different maximum datagram size limitations. MPTN architecture supports the maximum datagram size expected by the transport user.

Connectionless modes supported by MPTN architecture are:

- Unicast Datagrams

  The MPTN architecture allows a transport user to send a datagram to a single partner.

- Multicast Datagrams

  The MPTN architecture allows a transport user to send a datagram to a group address. The sender need not know which programs belong to the group and need not be a member of the group itself.

  Because multicast service is a significant service that is provided on various local area networks, it is included in the MPTN architecture. MPTN architecture support for multicast requires that all members of a multicast group have the same address, and thus have to be located in a common network.

- Broadcast Datagrams

  Broadcast service is treated as a special case of multicast. A broadcast datagram is a datagram sent to all transport users in a given network that have indicated a willingness to receive broadcast messages. Transport users that expect to use a broadcast function will join this broadcast group. The broadcast service available in the MPTN architecture provides distribution to the members of a single network, rather than all interconnected networks.

MPTN Access Node Function

This section describes the key functions provided by the the MPTN access node. This includes compensations as well as connection and termination.

Compensations: When the transport provider does not offer a service required by a transport user, the MPTN manager provides a compensation to fill the gap. A fixed collection of compensations is available to make up for all potential mismatches. The MPTN manager determines which compensations are needed for a particular connection and inserts headers as required.

Connection-Establishment and Termination: Establishment and termination of end-to-end transport-user connections are the two most sophisticated functions of the MPTN access node.

In connection establishment, the source transport user indicates the desire for a connection with its partner by providing the partner's transport-user address and requested service mode to its local MPTN managers. The source employs MPTN address mapping (which is described in the next section) to learn the partner's transport-provider address. It then uses MPTN connection-establishment protocol and the connection-establishment protocol of its selected transport provider to establish an MPTN connection with appropriate service characteristics to the indicated end user. In the process, it exchanges information about the characteristics of the end-to-end transport-user connection with its partner.

In the connection-termination process, the MPTN manager must ensure that the appropriate semantics are observed and then close the underlying transport connection.

MPTN Address-Mapping Function

In today's environment, each transport protocol has its own address space. For example, SNA uses LU names and TCP uses IP addresses. In a multiprotocol environment, an MPTN access node has two transport address spaces: the transport-user address space and the transport-provider address space, as shown in Figure 4 on page 12. To avoid disruption to either transport-users or transport-providers, MPTN allows a transport-user to use its existing transport address format, while the serving transport-provider uses transport addresses that its network expects.

MPTN must serve two needs related to transport addressing:

- Resolving potential transport-address conflicts among different protocols
- Mapping from the destination transport user address (the address a transport-user uses to address a partner transport user) to the corresponding destination transport-provider address (the address used in the underlying transport provider to address a partner transport provider)

The first need occurs because MPTN interconnects heterogeneous networks; therefore, MPTN must manage transport addresses of different formats, which might happen to use the same bit encoding.

The second need occurs because MPTN decouples a transport-user address from a transport-provider address; thus, a dynamic mapping between the two is required.

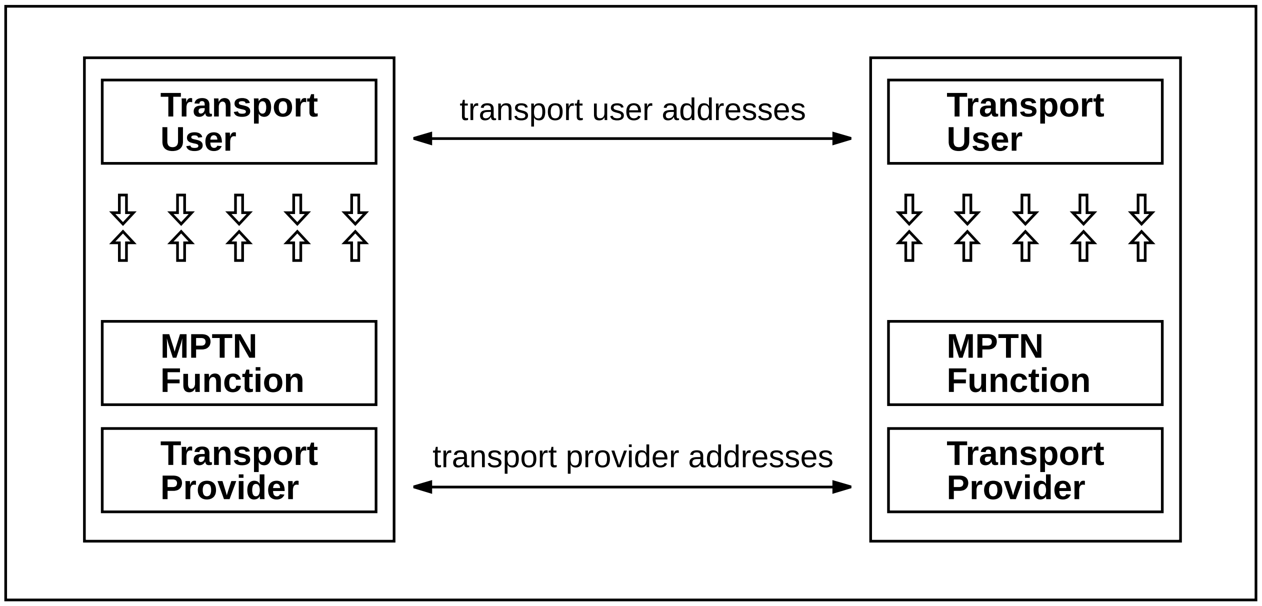


Figure 4. Transport Address Spaces

The MPTN architecture performs the following major address-mapping functions in order to map a transport-user address to a transport-provider address dynamically:

- A transport user registers its own address, making itself accessible to other transport users.
- The MPTN architecture maps a transport-user address to the corresponding transport-provider address and makes this mapping available to other MPTN nodes. This allows a user to set up a non-native connection with, or to send a non-native datagram to, another transport user.

There are three alternative mechanisms for MPTN address mapping:

1. Address mapping algorithms
2. Existing protocol-specific directories
3. Address mapper function, an MPTN component that maps transport-user addresses to transport-provider addresses

In all three cases, only node-level addresses are handled by the address mapping technique. Local program selectors such as TCP port numbers are managed strictly by the local nodes.

When MPTN transport gateways are involved, the mechanisms used in the source and destination networks do not have to match.

A network can support more than one mechanism for address mapping. The following sections "Algorithmic Mapping," "Protocol-Specific Directories," and "MPTN Address Mapper" describe how each alternative provides the two major address-resolution functions. Each has advantages and limitations, which are also described.

Algorithmic Mapping: A transport provider can implement an algorithm that generates a native address from the corresponding transport-user non-native address; thus, no database is required to hold the mappings. When transport-user addresses are registered, they are converted to the corresponding transport-provider addresses, which are then registered to the existing protocol-specific directory. An algorithm translates only between specific address types. Sockets over SNA uses algorithmic mapping, with IP host addresses mapped to SNA LU names.

When the addresses of non-native destination transport users are to be resolved, the algorithm is used by the source node to obtain the corresponding native addresses. Connections can then be built and datagrams delivered to the partners using protocol-specific mechanisms.

Algorithmic mapping is not always possible, because larger address spaces (such as SNA LU names which can be 16 characters) cannot be mapped one-to-one to smaller address spaces (for example, to a TCP address of 4 characters).

Protocol-Specific Directories: Address mapping can also be achieved by using existing protocol-specific directories to handle transport addresses of various formats. With this alternative, all transport-user addresses are registered with the protocol-specific directory. For example, to support a new address type on TCP/IP, a new domain can be added to the domain name system.

The limitation of using the protocol-specific directories is that not all directories support the registration of different address types and might not support dynamic updates.

MPTN Address Mapper: When neither the algorithmic mapping technique nor the protocol-specific directories technique applies, MPTN provides a component called the MPTN address mapper to provide the service. The address-mapper function can be placed in a node by itself, or in an MPTN access node, or in an MPTN transport gateway. SNA over IPX is an example in which a protocol pair requires the MPTN address-mapper function, because the larger SNA address space cannot be mapped onto the IPX addresses, and currently no IPX directory can be used for non-native address mapping.

The procedures carried out by an MPTN address mapper through datagram exchanges with its partners include:

- Registration

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

- Resolution

  A source or a gateway MPTN manager requests an address mapping from the MPTN address mapper when trying to bring up the next hop in a connection or to route a datagram. The MPTN address mapper can return to the requesting MPTN manager either the destination transport-provider address (if the two partners are connected to the same transport network) or the address of the first MPTN transport gateway.

MPTN Transport Gateway Function

The MPTN transport gateway is a major part of the MPTN architecture. It allows networks that have different network protocols to be concatenated so they appear to the user as one logical network.

The MPTN transport gateway takes care of resource location in the MPTN network, routes connections and datagrams through the MPTN network, sets up the connection through the MPTN network, and relays the data from one network to the next.

An MPTN transport gateway can be accessed by end systems that have no MPTN function (native nodes), as well as by MPTN access nodes. This allows applications that are running natively (for example, a socket application running over TCP/IP) to interoperate with a peer running non-natively (for example, a socket application running on an SNA network). MPTN gateways enable native nodes to access the global MPTN network.

MPTN Gateway Protocols: The MPTN transport gateway interacts with two types of nodes:

1. Native nodes
2. MPTN access nodes

Interactions with Native Nodes: When an MPTN gateway is attached to a native transport network with native nodes, the MPTN gateway accepts native protocol traffic as requests for MPTN services. This involves appearing as a full-function native node itself, including participating in the native routing protocols of that native protocol.

The MPTN gateway must function as a native router for that protocol. This involves advertising its ability to reach end systems outside the native transport network, and the ability to learn how to route to end systems within the native transport network. The MPTN gateway must also be able to receive all incoming native searches and respond when the requested end system exists outside the native transport network. The MPTN gateway must also be able to initiate a search in the native transport network when a connection or datagram is sent from another native transport network.

Interactions with MPTN Access Nodes: MPTN gateways interact with other access nodes using the same MPTN protocols that access nodes use to interact with each other.

MPTN Network Management

In an MPTN environment, individual single-protocol transport networks are managed by a resource-specific manager that understands the appropriate native management protocol for that native transport network, whether it is SNA Management Services (SNA/MS), Simple Network Management Protocol (SNMP), or Common Management Information Protocol (CMIP). The additional function required by MPTN is the ability to correlate the transport-user address with that of its transport provider, and the correlation of the two connections in an MPTN transport gateway.

With MPTN transport gateways, connections and datagram routing paths can traverse multiple native transport networks. Managing multiple-protocol connections and datagrams requires establishing transformations from the single-protocol network management to the connections and datagrams seen by the transport user.

Network Services

Today there is a plethora of different transport networking protocols. Each protocol has its own advantages and disadvantages in terms of network performance and overhead, ease of network management, availability of important application programs, presence of implementations by multiple vendors, and congestion-control characteristics. Each has its proponents. Until recently, network owners selected a networking protocol and stuck to it. This is no longer true. With the availability of transport-independent interfaces, we are able to support the protocol of the customer's choice.

The structural separation of transport services and subnetworking services allows customers to separate the choice of networking technologies and protocols from the choice of transmission technologies (wireless, ATM, frame relay, and ISDN), and optimize each decision on its own merits.

Data communication requires that network components agree on both the layouts of the messages they exchange and the actions they take based on the kinds of data they receive. The layouts are referred to as *formats*, and the actions taken are referred to as *protocols*. This section briefly reviews some of the major protocols.

TCP/IP Protocols

The term *TCP/IP* refers to a family of standards-based network protocols, of which TCP, providing host-to-host connections, and IP, providing data routing from source to destination, are two important parts. TCP/IP protocols allow different networks to function as a single coordinated entity. TCP/IP was originally developed to link U.S. Government, military, research, and university networks. TCP/IP is not vendor-specific and has been implemented on everything from personal computers to the largest supercomputers. It is used for both LANs and WANs and by many different government agencies. Today these networks, together with many commercial networks connected with them, are collectively referred to as the *Internet*.

It is important to realize that while sites on the Internet use the TCP/IP protocols, many other organizations have established their own networks using the same TCP/IP protocols.

Basic TCP/IP Architecture: TCP/IP consists of a four-layered structure of protocols (see Figure 5) ranging from low-level, hardware-dependent programs, to high-level applications. We will focus on the functions of the transport and internetwork layers. Access to the transport and internetwork layer functions in many operating environments is accomplished through an API known as *sockets*. The sockets API originated as a general interprocess communications facility of Berkeley software distribution (BSD) Unix systems and is still prevalent in today's BSD-derivative Unix systems. Further, the TCP/IP-relevant components of the sockets API were adopted as the foundation for TCP/IP interfaces on non-Unix platforms such as OS/2 (OS/2 TCP/IP sockets) and Windows (Winsock).

Transmission Control Protocol (TCP): The Transmission Control Protocol (TCP) provides reliable delivery of a stream of bytes in sequence between hosts in an internet. TCP takes a stream of data, breaks it into segments (a TCP header and application data), sends each one individually (using IP) and then reassembles the segments into the original stream. If any segments are lost or damaged during transmission, TCP resends the missing segments.

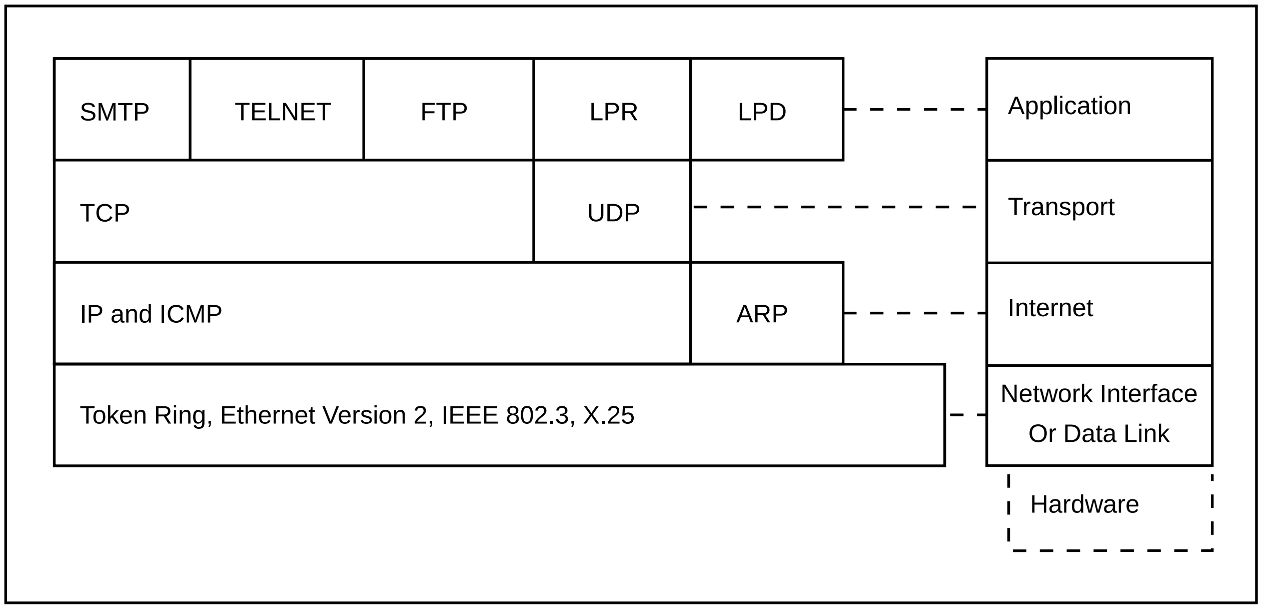

SMTP
TELNET
FTP
LPR
LPD
Application
TCP
UDP
Transport
IP and ICMP
ARP
Internet
Token Ring, Ethernet Version 2, IEEE 802.3, X.25
Network Interface Or Data Link
Hardware

Figure 5. The TCP/IP Layered Architecture

User Datagram Protocol (UDP): The User Datagram Protocol (UDP) allows application programs to send datagrams to programs on other systems with a minimum of protocol overhead. Unlike TCP, UDP does not offer guaranteed delivery of data. UDP can be used instead of TCP when the overhead of TCP connecting and disconnecting is not desired, when the application will do its own acknowledgment and retransmitting to ensure reliable data transfer, or when retransmission is not required.

Internet Protocol (IP): The Internet Protocol (IP) provides the basic transportation rules for communication between hosts on the different networks that make up the Internet. A host on the network has a unique internet address and an associated system name. IP is responsible for routing packets from a host on one network through a series of routers to a host on another network.

In IP-based networks, information is transmitted between hosts in the form of datagrams. A datagram includes an IP header and data. A datagram is the basic unit of information in TCP/IP, consisting of a source address, destination address, and data. At the Internet level, all addressing is host-to-host, using fixed-length addresses to identify source and destination hosts. The protocol layers above only need to know each host's Internet address to make a connection.

Internet Control Message Protocol (ICMP): The Internet Control Message Protocol (ICMP) provides for error and control messages between host systems (peer computers in a network) and routers. Routers and host systems use ICMP to send reports of problems. ICMP also includes an echo request or reply used to test whether a destination can be reached and can respond (the PING command).

Address Resolution Protocol (ARP): The Address Resolution Protocol (ARP) dynamically locates the hardware address of a given IP address. ARP relies on the broadcast capabilities of the underlying media (Token Ring, X.25, and so on) to provide this function.

Emerging TCP/IP Network Infrastructure: The basic TCP/IP architecture has shortcomings with respect to some important requirements envisioned for the Internet and large private IP networks. The Internet Engineering Task Force (IETF) is the standards-making body that develops new standards for TCP/IP internetworking. Following are notable issues under discussion in the IETF regarding enhancements to TCP/IP transport services and networking infrastructure.

Automatic Host Configuration: Traditionally, each host in a TCP/IP network must be manually configured with a set of TCP/IP-related networking parameters to gain access to an IP network. These parameters, which consist of things such as an IP address and local routing and service information, are typically provided by an administrative authority, which must track the associations between users, hosts, IP addresses, routers, and other network services. Further, these associations can vary according to user classification or host location.

In large networks, then, the tasks performed by TCP/IP network administrators are large, cumbersome, and error-prone. A new protocol called the Dynamic Host Configuration Protocol (DHCP) eases IP host configuration and network administration. DHCP is a client/server protocol whereby a DHCP client host dynamically learns its IP address and other network configuration parameters from a DHCP server in the network.

Also, a complementary standard to DHCP called Dynamic DNS is currently under development. Dynamic DNS enables hosts to update TCP/IP well-known names, called *domain names*, in the Domain Name System (DNS). For example, a DHCP client can update its name-to-IP-address mapping in DNS to reflect the new IP address it obtained from a DHCP server.

IP Mobility: The IP addressing structure and routing architecture are designed for routing efficiency and assume that a host's address is associated with its physical location within an internet. That is, if a host physically moves from one part of an internet to another, its address must change to reflect its new location. This implies that maintaining a host's IP connectivity while it changes locations within an internet is fundamentally impossible without further redesign of the basic TCP/IP protocols.

Mobile IP is an emerging standard that enables a mobile host to maintain its IP address independent of its physical location. This allows a host to maintain its connectivity, including established TCP connections, as it moves about. To accomplish this seamless connectivity, Mobile IP proposes new internet routing elements that track the location of a mobile host and forward packets to it wherever it roams.

If seamless connectivity while moving about is not a requirement, DHCP and DDNS (together known as Dynamic IP) provide mobility by enabling hosts to automatically configure themselves for network access from any point on the internetwork (refer to “Automatic Host Configuration” on page 16). Dynamic IP hosts can, therefore, relocate anywhere in an internetwork without reconfiguring. Unlike hosts using Mobile IP, however, Dynamic IP hosts can have to re-establish their network connections when they move to a new location in the internetwork.

Integrated Services Internet: The Internet is envisioned as being a single, high-speed network that integrates the transport of all types of electronic information, including time-critical interactive audio/video data as well as traditional interactive and bulk data. The current TCP/IP architecture and routing mechanisms do not provide the means to differentiate between these data types or the levels of service they require.

Resource Reservation Setup Protocol (RSVP) is an emerging standard that defines mechanisms to allow applications to request levels of service from the transport and routing infrastructure. This enables, for example, network applications requiring high-speed, low-delay network transmission to reserve enough bandwidth to be usable, independent of surges in network traffic from other sources.

Another proposed standard called Real-Time Protocol (RTP) uses adaptive network performance techniques to circumvent the lack of service guarantees in today's IP networks and provide an end-to-end transport service suitable for real-time data such as audio and video.

Secure IP Transport: TCP/IP currently has no inherent mechanisms to authenticate partner hosts or to ensure that transmissions between hosts are private. A set of standards collectively known as IPSEC fixes those shortcomings to enable any TCP/IP application to benefit from secure networking (for example, file transfer of confidential data over the public Internet). IPSEC defines mechanisms for authenticating IP packets and for encrypting IP data payloads. In addition, IPSEC defines key management protocols which enable two hosts to negotiate these mechanisms and associated security credentials to establish secure communications.

In the interim, a de facto standard called Secure Sockets Layer (SSL) is being used to provide a level of security for internet applications, such as private transactions over the Internet. SSL implements a layer of data security between application protocols, (such as HTTP, Telnet, FTP, and so on) and TCP/IP. SSL provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection.

SSL provides a security "handshake" that is used to initiate the TCP/IP connection. This handshake results in the client and server agreeing on the level of security they will use. Thereafter, the role of SSL is to encrypt and decrypt the data of the application protocol being used.

Because of the popularity of the World Wide Web (WWW), SSL is often compared to SHTTP (Secure HyperText Transfer Protocol) because both can be used to secure WWW transactions. Although similar in purpose, SSL and SHTTP are very different in implementation: SSL is a general security layer that can be used beneath many applications, whereas SHTTP is a security-enhanced variant of the HTTP protocol of the WWW that adds message-based security.

It is important to note that SSL and SHTTP are not mutually exclusive, and they easily coexist by layering SHTTP on top of SSL.

IPv6: "The Next Generation": IPv6, or IP "Next Generation" protocol, addresses the shortcomings of today's TCP/IP (IPv4) protocol, including the projected address space shortage accelerated by the popularity of the Internet. Other traditional architectural deficiencies addressed include network security, automatic address assignment and host configuration, and transport levels of service.

In IPv6, addresses are 128 bits in length versus the current 32 bits, and the new addressing structure and network protocols facilitate migration to IPv6 from environments with existing IPv4 networks and hosts.

The SNA Protocols

Systems Network Architecture (SNA) is a data communication architecture established by IBM to specify common conventions for communication among the wide array of IBM and non-IBM hardware and software data communication products. The manner in which products internally implement these common conventions can differ from one product to another. Because the external *interface* of each implementation is compatible, different products can communicate without the need to distinguish among the many possible product implementations. The specification of SNA is well-defined and publicly available. SNA implementations are available from a large number of vendors, and they work well in an open, heterogeneous environment.

The following are some examples of hardware components that support SNA transport network functions:

- Processors such as the ES/9000 family
- Distributed processors such as the Application System/400
- Communication controllers such as the 372X and 374X series
- Cluster controllers
- Workstations

The following are some examples of software components that implement SNA transport network functions:

- Telecommunication access methods such as the Virtual Telecommunications Access Method (VTAM) and Communications Manager/2 (CM/2)
- Network control programs such as the Advanced Communication Function for Network Control Program (NCP)

SNA consists of a seven-layered structure of protocols ranging from low-level, physical control to high-level transaction and application services. This section describes the major functions of the transport and networking layers.

Maximize Network Throughput: When too much data is introduced into the network, congestion can occur. Severe congestion blocks the flow of messages and can result in the loss of data or inefficient retransmissions. SNA provides end-to-end controls to prevent overload and deadlocks by controlling the pace at which end-to-end data flows through the network. Therefore, the end partners can adapt to network conditions to maximize the throughput of the entire network.

Reduced Definitions: Partners perform dynamic negotiation of their capabilities to determine the highest level of capability. Thus, the manual system definitions or customizations can be reduced, allowing better use of the network and a smaller operations staff. If one partner upgrades its capability, the other partner will get the new higher capability without any user intervention.

Network Security: The need for data security is increasing in today's networks because they are the vehicles for such sensitive data as banking transactions and personal information. SNA provides data encryption, authentication, and access control as part of network security. Protocols for data encryption and password verification meet the data security objective of restricting access to authorized users or systems and limiting visibility of data.

Network Priority: SNA provides fair use of network services to network users. Transmission priority and class of service enable preferential treatment for different types of network traffic. For example, interactive traffic can be given a higher priority than file transfers, which are not time critical.

Route Selection: Path control can select optimal routes for data transmission to match customers' requirements for security and response time. With this type of route selection service, all links can be efficiently utilized regardless of their speed and error characteristics.

Data Segmentation: Data segmentation optimizes the use of network capacity by improving the link utilization for links with large error rates and increasing the throughput for nodes with limited buffer space.

Advanced Peer-to-Peer Networking Protocols

Advanced Peer-to-Peer Networking (APPN) is an enhancement to SNA to dynamically tie together diverse platforms and topologies of rapidly changing networks into a single network. APPN's any-to-any connectivity makes it possible for large and small networks alike to communicate over local and wide area networks, across slow and fast links.

APPN adds two key functions beyond SNA that makes APPN easy to install and operate. It dynamically keeps track of the location of resources in the network, and it selects the best path to route data between resources. APPN nodes dynamically exchange information about each other.

APPN features help eliminate unnecessary network control traffic, thus providing more bandwidth for moving data through the network. In APPN, only routing nodes exchange topology information, and they exchange this information only when changes occur in the backbone of the network. APPN nodes limit searches for other resources in the network, and, as a result, improve network performance.

APPN's priority service ensures that important data moves through the network quickly. By using intelligent class-of-service routing, APPN nodes consider factors such as security, cost, delay, and throughput to select the best route for different types of data. The APPN protocol avoids network congestion by using adaptive pacing, which ensures more consistent response time.

APPN protocols reduce the negative impact of a failure of an individual component in the system by decentralizing control of the network and selecting routes based only on the current status of the network topology, not on predefined static information. APPN includes network topology updates and automatic route selection, which make it unnecessary to predefine node locations and routes between nodes.

High-Performance Routing (HPR) is an evolutionary extension to APPN. It enhances data routing performance by decreasing intermediate node processing and taking advantage of the higher speeds and reliability of today's links. HPR also increases session reliability with “nondisruptive path switch” that switches sessions automatically to reroute data around a failed node or link without disrupting the application connections. With the high-speed transmission media, it is unnecessary to run a full recovery DLC protocol between each pair of SNA nodes. Instead, the few frames in error are simply discarded by the node detecting the error and HPR recovers the frames on an end-to-end basis. Furthermore, HPR uses selective retransmission to greatly increase the throughput when errors are encountered.

HPR uses adaptive congestion control to more intelligently manage network congestion in contrast to congestion control techniques in other protocols that automatically discard data upon detecting congestion. Discarding data causes retransmission and can in fact exacerbate the congestion condition.

Network Basic Input/Output System (NetBIOS) Protocols

NetBIOS is one of the commonly used protocols for the LAN environment today. It is supported on the Ethernet, token ring, and IBM PC Network environment. IBM introduced NetBIOS originally for use on the PC Network, an early LAN. It was defined as an interface between the application programs and the network adapter LANs. However, some transport-like functions have been added. It is one of the de facto standards in the LAN environment, although several variations of NetBIOS are used.

NetBIOS was designed for a group of personal computers, all sharing a common broadcast medium. It provides both a connection-oriented (virtual circuit) and a connectionless (datagram) service. It supports both broadcast and multicast. Three types of services are provided by NetBIOS:

- Name service
- Session service
- Datagram service

NetBIOS provides functions of session, transport, and network services. Due to its minimal network-layer functions and simple transport layer, NetBIOS is used primarily in the LAN environment. NetBIOS primarily uses the 802.2 interface to provide datagram or session layer communication between applications executing on different machines in the network.

Name Service: Names are used to identify resources in NetBIOS. For example, for two processes to participate in a conversation, each must have a name. The names are dynamically assigned and checked. The client process identifies the specific server by the server's name, and the server can determine the name of the client. There are two types of names: unique names and group names. A unique name must be unique across the network. A group name does not have to be unique, and all processes that have a given group name belong to the group.

Each NetBIOS node maintains a table of all names currently owned by that node. These names continue to be owned by the NetBIOS node until the names are specifically deleted or until the node is powered off or reset.

Session Service: The NetBIOS session service provides a connection-oriented, reliable, full-duplex message service to a user process. NetBIOS does not provide any form of out-of-band data. NetBIOS requires one process to be the client and the other to be the server.

NetBIOS session establishment requires a preordained cooperation between the two stations. One application must have issued a Listen command when another application issues a Call command. The Listen command references a name in its NetBIOS name table and the remote name an application must use to qualify as a session partner. If the receiver (listener) is not already listening, the Call command will be unsuccessful. If the call is successful, each application receives notification of session establishment with the session ID. The Send and Receive commands then transfer data. At the end of a session, either application can issue a Hang-Up command. There is no real flow control for the session service, because it is assumed a LAN is fast enough to carry the required traffic.

Datagram Service: Datagrams can be sent to a specific name, sent to all members of a group, or broadcast to the entire LAN. As with other datagram services, such as UPD/IP, the NetBIOS datagrams are connectionless and unreliable. The Send_Datagram command requires the caller to specify the name of the destination. If the destination is a group name, then every member of the group receives the datagram. The caller of the Receive_Datagram command must specify the local name for which it wants to receive datagrams. The Receive_Datagram command also returns the name of the sender, in addition to the actual datagram data. If NetBIOS receives a datagram, but there are no Receive_Datagram commands pending, then the datagram is discarded.

The Send_Broadcast_Datagram command sends the message to every NetBIOS system on the local network. When a broadcast datagram is received by a NetBIOS node, every process that has issued a Receive_Broadcast_Datagram command receives the datagram. If none of these commands are outstanding when the broadcast datagram is received, the datagram is discarded.

IPX/SPX Protocols

Internetwork Packet Exchange (IPX) Protocol: The functions of IPX are equivalent to the combined functions of IP and UDP. Through IPX, a workstation can send or receive packets to or from any node on the internetwork. The routing of packets between nodes on different LANs is transparent to the applications on the workstation. The network drivers make a best effort to deliver packets but do not guarantee delivery.

IPX is a connectionless, datagram protocol. Thus IPX packets containing data are addressed and sent to their destinations, but there is no guarantee or verification of successful delivery. This packet acknowledgement, or connection control, must be provided by protocols above IPX (SPX).

The tasks performed by IPX include addressing, routing, and switching information packets to move packets from one location to another on the network. IPX defines internetwork and intranode addressing schemes. Network numbers are the basis of IPX's internetwork addressing. Each network segment on an internetwork must be assigned a unique network number, which is used by routers to forward packets to their final destination segment. The addressing of the packet is handled in its MAC header and IPX header address fields. The MAC header contains the node addresses of the router and the source. The IPX header contains the internetwork addresses of the source and the destination. The IPX intranode address comes in the form of socket numbers, which provide a quick method of routing packets within a node.

Routers for IPX: Routers interconnect different network segments and, by definition, are network layer devices. IPX performs these network layer protocol tasks for routers (IPX performs tasks outside the network layer as well). IPX routers can be configured to interconnect two or more segments. Each of these segments, however, must be assigned a unique IPX network number to distinguish it from other segments on the internetwork. The network number serves as a common address for each node connected to a segment.

Packet Delivery: On an IPX network, the successful delivery of a packet depends on the proper addressing of the packet and the internetwork configuration. The addressing of the packet is handled in its media-access protocol header and IPX header address fields. To send a packet to another node, the sending node must know the full internetwork address (network, node, and socket) of the node it desires to send to.

Sending Node's Responsibility: When a node wants to send information to another node with the same network number (both nodes are on the same segment), the sending node can simply address and send packets directly to the destination node. However, if the two nodes have different network numbers (reside on different network segments), the sending node must find a router on its own segment that can forward packets to the destination node's network segment. To find this router, a workstation broadcasts a routing information protocol (RIP) packet requesting the fastest route to the destination node's network number. The router residing on the sending node's segment with the shortest path to the desired segment responds to the RIP request. When the sending node is a router rather than a workstation, the router can get this information from its internal routing tables and need not send the RIP request. Once the sending node knows the router's node address, it is prepared to send packets to the destination node.

Router's Responsibility: When a router receives an IPX packet, its IPX handling process determines how to route the packet. If the packet is addressed to the router, it should be handled internally by the appropriate socket process, otherwise further routing will be required. If, however, the router is not directly connected to the segment that the final destination node resides on, it will send the packet to the next router in the path to the destination node.

Mixed Topology Routing: The routers should take advantage of IPX enhancements to route packets of any size. A problem arises when a large packet from one network topology requires routing to another topology that cannot handle the large packet. In this case, the packet should simply be discarded. It is the responsibility of the sending node, not the router, to make sure that the packet is sized such that it can be handled at the destination node.

Sequenced Packet Exchange (SPX) Protocol: The SPX protocol is built upon the IPX packet protocol to provide guaranteed delivery. The SPX protocol enables two communication client processes to exchange messages in a sequenced packet stream with guaranteed packet delivery and duplicate packet suppression.

The SPX interface provides a guaranteed delivery packet stream by using IPX datagram primitives to send packets and receive positive acknowledgement of packet delivery. Packets that are not acknowledged within a specified time interval are retransmitted by SPX. After a reasonable number of retransmissions have failed to return a positive acknowledgement, the connection is assumed to have failed.

SPX provides a time-out scheme to guarantee recovery from failed requests to establish a connection, send sequenced packets, or terminate the connection. Appropriate timeout values are selected by SPX to match the physical characteristics of the intervening networks.

SPX is connection-based rather than socket-based. This allows easier matching of received packets to listen from concurrent connection-based and socket-based clients.

Appendix A. Bibliography

Open Blueprint Technical Overview, GC23-3808.

Multiprotocol Transport Networking (MPTN) Architecture: Technical Overview, GC31-7073.

Multiprotocol Transport Networking (MPTN) Architecture: Formats, GC31-7074.

Multiprotocol Transport Networking (MPTN) Architecture: Tutorial and Product Implementations, GG24-4170.

Systems Network Architecture Technical Overview, GC30-3073.

TCP/IP Tutorial and Technical Overview, GG24-3376.

UNIX Network Programming by W. Richard Stevens, Prentice Hall Software Series.

Communications for Cooperating Systems by R. J. Cypser, Addison-Wesley Publishing Company.

Appendix B. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced-Peer-to-Peer Networking
AnyNet
Application System/400
APPN
ES/9000
IBM
IBMLink
Open Blueprint
OS/2
VTAM

The following terms are trademarks of other companies:

DCE	The Open Software Foundation
Internet Packet Exchange	Novell, Incorporated
IPX	Novell, Incorporated
X/Open	X/Open Company Limited

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

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

Appendix C. Communicating Your Comments to IBM

If you especially like or dislike anything about this paper, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply. Feel free to comment on specific error or omissions, accuracy, organization, subject matter, or completeness of this paper.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

- If you prefer to send comments by FAX, use this number:

  United States and Canada: 1-800-227-5088.

- If you prefer to send comments electronically, use one of these ID's:
  - Internet: **USIB2HPD@VNET.IBM.COM**
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**

Make sure to include the following in your note:

- Title of this paper
- Page number or topic to which your comment applies

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC23-3932-01