SNA Perspective

Volume 14, Number 4
April, 1993
ISSN 0270-7284

The single source, objective monthly newsletter covering IBM's Systems Network Architecture

IBM Starts Rollout of Multiprotocol Transport Networking

Translation is always a challenge, whether in human dialogue or data communications. Most companies today find themselves with multiple application environments and protocols for a variety of reasons ranging from independent departmental decisions to mergers and acquisitions. Several approaches have been proposed over the years to address the need for multiprotocol transport. The user can select from one or more options to address the challenge of multiprotocol environments. These options include:

- Limit the network to a single API/protocol/LAN-WAN stack and replace all applications and systems that do not and cannot conform
- Bridge or encapsulate as needed between different protocol environments
- Choose one general (many-to-many) or several specific (one-to-one) conversion approaches at one of several levels—network, transport, API, or application level

Unveiled in 1992 as a component of the networking blueprint, IBM's multiprotocol transport networking (MPTN) is the latest proposal for a general approach to addressing the complexity of this issue. In March, IBM announced products which implement the MPTN architecture. Not intending MPTN to be a proprietary approach, the company is actively soliciting vendor partners to implement MPTN and is working with standards bodies for MPTN standardization.

This article describes MPTN and its place in IBM's networking blueprint, discusses current and expected MPTN products, compares MPTN to several other strategies and products for multiprotocol transport including data link switching, tn3270, APPN over sockets, the CICS-sockets interface, and APPI, and considers issues and concerns regarding the MPTN approach.

In This Issue:

IBM Starts Rollout of Multiprotocol Transport Networking 1

The first products based on the multiprotocol transport networking architecture are soon to ship. Because you have told us how important network flexibility is to you and because we consider MPTN to be the heart of the networking blueprint, we have devoted this entire issue to explaining MPTN—what it is and what it isn't. Transport compensation and address mapping are two intriguing elements we explore. We also compare it with other multiprotocol transport products such as tn3270, data link switching, and even APPI. We also go beyond IBM's formal statements of direction to posit the third wave of MPTN implementations.

Architect's Corner: Mixing Oil and Water 14

One of the hottest topics in networking today is intermixing protocol environments—over, under, across, and through. Our architect is unsettled by the myriad of combinations and the variety of solutions. But while the world is not simpler, at least we have options.

(continued from page 1)

MPTN Background

MPTN, Common Transport Semantics, and the Networking Blueprint

MPTN is part of the broader common transport semantics (CTS) in IBM's networking blueprint. The networking blueprint, as shown in Figure 1, is IBM's framework for structuring the enterprise networking environment. *SNA Perspective* discussed the networking blueprint in the August 1992 article "Blueprint to Integrate the Architectures."

As indicated in the figure, CTS represents a common means for interfacing several applications or application programming interfaces (APIs) to a variety of transport protocols. MPTN, not shown explicitly in the figure, is one of the means to provide CTS. There are two main benefits of CTS and MPTN:

- Application independence
- Network simplification

Application independence with CTS is intended to decouple applications from dependence on a particular type of transport so that the user can select an application based on its own merits and not on its ability to run on the user's existing network. The network simplification benefits allow the user to install and manage fewer protocols and even to eliminate parallel networks.

CTS Includes RFCs

Currently, in addition to MPTN, CTS includes Requests for Comments (RFCs) 1001, 1002, and 1006 of the Internet Engineering Task Force (IETF). RFC 1006 supports OSI applications running over transmission control protocol/internet protocol (TCP/IP) transport. RFC 1006 is also known as the International Standards Organization Development Environment (ISODE) and was developed in the mid1980s to support OSI application development even if a user did not have an OSI network installed. RFC 1001 and 1002 describe NetBIOS applications running over TCP/IP. CTS may eventually include additional standards as they emerge.

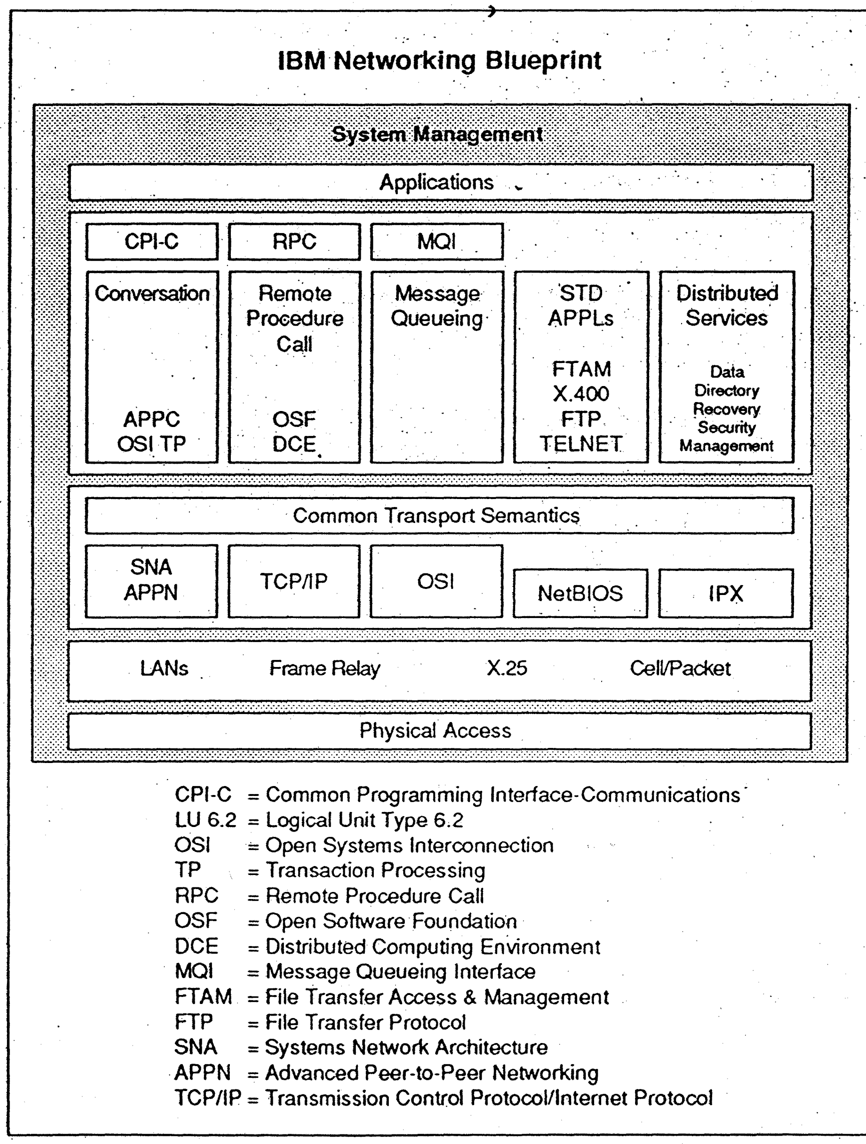

Figure 1

MPTN, XTI, and X/Open

MPTN has been proposed by IBM as one of several extensions to an interface developed by X/Open called the X/Open Transport Interface (XTI).

XTI was initially developed as a common interface for OSI and TCP/IP environments and has been extended to include NetBIOS. XTI provides mapping of similar capabilities but not compensations for differing capabilities. XTI defines a common interface but requires the application to specify which transport is to be used. XTI also requires that both partners be attached to the same transport protocol.

IBM's proposal builds upon XTI in three ways:

- Extends the supported protocols to include SNA
- Provides transport compensations for dissimilar capabilities between protocols and allows the application to be unaware of the actual transport selected
- Allows partners to be attached to different transport types by defining gateway nodes as well as access nodes

The first extension to XTI listed above is not part of MPTN. IBM also proposed some other detailed technical additions to XTI. X/Open seems more interested in some elements of IBM's proposal than others. Below in this article, we discuss the status of IBM's submission to X/Open.

How MPTN Works

Only for Matching APIs

MPTN supports communication between two matching application types or APIs, such as two CPI-C applications or two sockets applications, but not between different API types, such as a CPI-C application with a sockets application (see Figure 2).

Transport User and Provider

MPTN uses two terms in a specific way—user and provider. The transport *user* is the application or API which expects to run over a particular transport, as shown in Figure 3. The transport *provider* is the actual transport which will support the user. MPTN provides mapping and compensation between the transport needs of the user and the capabilities of the transport provider.

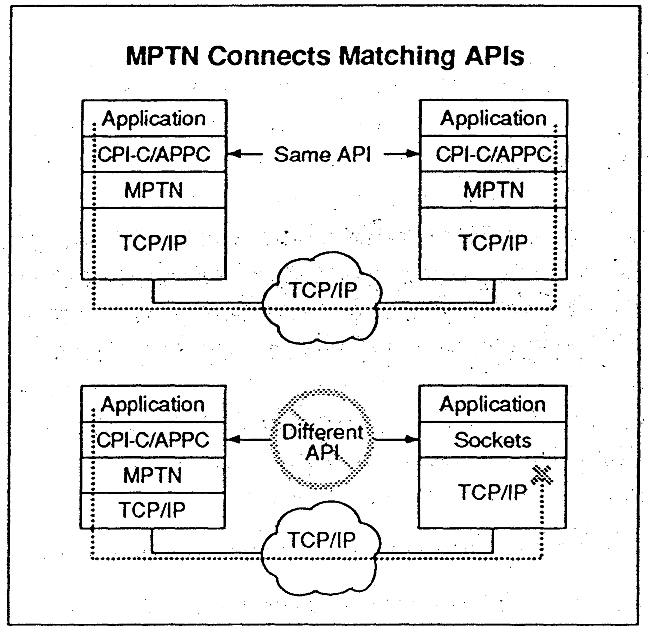

Figure 2

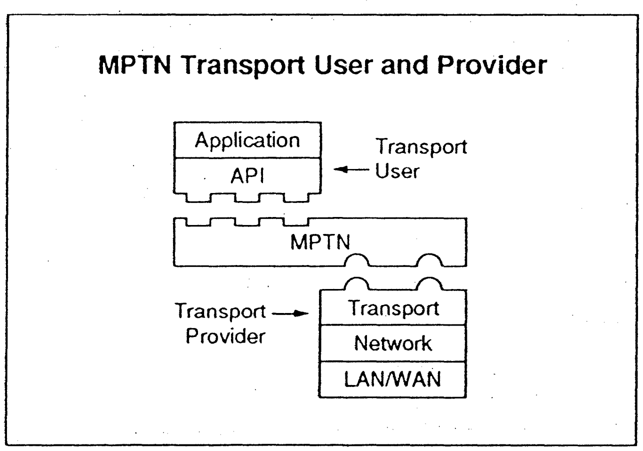

Figure 3

Access Node or Gateway Implementation

MPTN can be implemented in two ways—as an access node or gateway. In an *access node,* MPTN is in the same system as the application. One type of access node could be a server which supports access from clients on several transport types through MPTN.

In *gateway* mode, MPTN is located in a separate device from the applications, as shown in Figure 4. Note that the two networks in a gateway implementation do not have to be physically separate. The three devices shown in the gateway configuration, for example, could all be on a single token ring segment.

Transport Mapping and Compensation

MPTN, like XTI, describes a common interface between transport types. It supports mapping between two similar capabilities in the different protocols, such as two different ways to describe record boundaries.

However, unlike XTI, MPTN additionally provides compensation between transports with varying capabilities. For example, an SNA application sends data in record format while TCP/IP operates in a stream mode. Another example is that a TCP/IP application may send a multicast message (to more than one recipient) which SNA transport does not natively support. MPTN provides compensations for such requirements. Other examples where compensation is required are shown in Table 1 (see page 5). IBM claims that compensations are usually minimal and are included in the protocol header or a small MPTN header.

Three Techniques for Multiprotocol Transport

MPTN differs from two other approaches to multiprotocol transport which are shown in Figure 5—encapsulation and conversion. With most protocols, the application or API provides a header and the transport provides a header. For example, SNA applications provide a request/response header (RH) and a transmission header (TH) while an RPC application running over TCP/IP would use an RPC header and a TCP header. The three approaches differ in how they handle these headers.

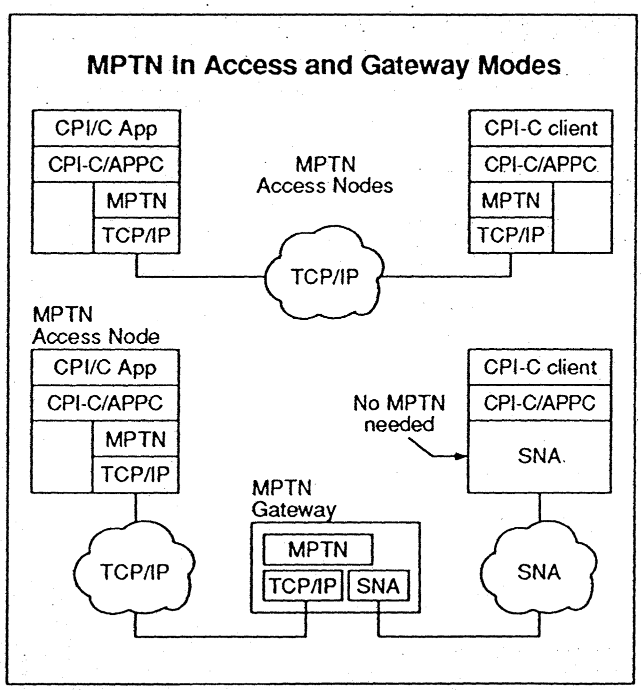

Figure 4

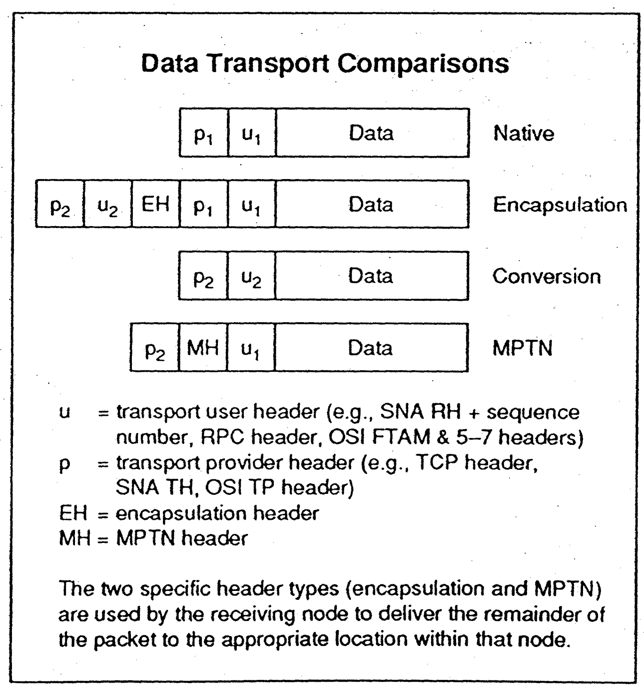

Figure 5

Encapsulation. Most multiprotocol routers today use encapsulation for transporting other protocols. In this case, each packet contains both sets of headers, one hidden or enveloped inside the other. This procedure is the most simple. However, it is also less flexible, usually requiring predefined destinations and/or routes. It also uses more overhead to send and process both sets of headers and control traffic. With more experience, encapsulation products are becoming more sophisticated and efficient. For example, many multiprotocol routers encapsulating SNA over TCP/IP use some form of poll spoofing to avoid sending frequent SNA polls.

Conversion. With a conversion approach, the packet is completely converted into another transport type. An example is IBM's CICS-sockets interface. With this interface, the application itself is changed to add the capability to communicate natively over a sockets connection. Conversion can also be done in a separate gateway device. When communicating over TCP/IP with conversion, no SNA protocols are sent over the TCP/IP portion of the network and the TCP/IP traffic may or may not be reconverted into SNA at the destination. Conversion approaches are more complex than encapsulation and are usually designed for a specific application such as the CICS-sockets interface or pair of applications or the SMTP-PROFS electronic mail gateway.

Transport Services Supported for Nonnative Transport Users

	SNA	TCP/IP	OSI	NetBIOS
Connection data	Compensation required	Compensation required	Compensation required if > 32 bytes	Compensation required
Termination data	Compensation required	Compensation required	Compensation required if > 64 bytes	Compensation required
Record boundaries	Supported	Compensation required	Supported	Supported
Expedited data	Supported[1]	Compensation required[2]	Compensation required if > 16 bytes	Compensation required
Correlation of expedited and normal data	Compensation required	Supported	Compensation required	Compensation required
Maximum size of expedited data	86 bytes[1]	1 byte	16 bytes	Expedited data not supported
Maximum size of connectionless data	Compensation required	Implementation dependent	Defined by underlying Network layer	Implementation dependent
Maximum record data	No limit	No limit	Defined by underlying Network layer	128 Kbytes
Orderly termination	Supported	Supported	Compensation required	Supported
Abortive termination	Compensation required	Compensation required	Supported	Compensation required
Simplex termination	Supported	Supported	Compensation required	Supported
Duplex termination	Compensation required	Compensation required	Supported	Supported
Connection outage notification	Supported	Compensation required	Supported	Compensation required

Source: IBM Corporation

Table 1

Cisco's Advanced Peer-to-Peer Internetworking (APPI) will combine these two approaches. APPI will convert the SNA control information, including directory and topology services, into TCP/IP formats but the LU-LU traffic will be encapsulated across TCP/IP.

MPTN. MPTN can be seen as operating between encapsulation and conversion. The application header is unchanged but the transport header is changed. With MPTN, the application believes that its traffic is being sent on its native transport. MPTN offers more flexibility than simple encapsulation and is more easily generalized than most conversion approaches.

The Case of SNA

SNA Applications. For SNA applications, MPTN performs an interesting twist on layer splitting. To describe it, we must first review how SNA works today. An SNA path information unit (PIU), sometimes called a frame or packet, includes a transmission header (TH) which is added by the transmission control layer and a request/response header (RH) which is added by the session layer.

The PIU sequence number, which is calculated by the session layer, logically belongs in the RH. However, in all SNA implementations, the sequence number is passed down to the transmission control layer and included in the TH instead. This has presented a significant problem for SNA layer splitting for multiprotocol solutions, since replacing the TH would remove the sequence number for ordering packets at the destination.

MPTN takes the sequence number generated in the session layer and prepends it to the RH before calculating an alternate header to replace the TH over the different transport.

SNA Transport. For SNA transport, MPTN operates over LU 6.2, not directly over APPN or subarea SNA path control. This is because the SNA LU function spreads over layers 4 through 6, which makes it more difficult to map cleanly to transport models that have more strictly layered modules. A benefit of using LU 6.2, however, is that MPTN traffic can run over subarea SNA as well as APPN, at least wherever the subarea supports LU 6.2 sessions.

Three Types of Address Mapping

In addition to handling the protocols themselves, a multiprotocol solution must deal with different address formats in the different environments, such as the association between a node's SNA address and its TCP/IP address. This is necessary to allow an application to specify a destination address in its native format and to find and access that device across a different transport.

The association between the two addresses may be preconfigured, as is often done in encapsulation solutions. Alternatively, there are several ways to dynamically map addresses.

MPTN uses one of three techniques for address mapping, depending on the degree of difference between the two address schemes in use.

- **Algorithmic.** The algorithmic approach is used when the user address space can map into the provider's address space, such as from an Internet user address to an SNA provider name. The transport provider uses the algorithm to calculate a transport provider address from the transport user address.
- **Extended Native Directory.** An extended native directory is used when the provider's directory supports registration of different address types. An example is that an APPN address with a network identifier and LU name (NETID.LUname) for a particular company can be registered on an Internet domain name server as LUname.NETID.SNA.companyname.com. In this way, APPN addresses can be registered on TCP/IP domain name servers.

- **Address Mapper.** The third technique, the address mapper, would only be used when the other two approaches are not feasible. At least one node in an MPTN environment will include an MPTN mapper which keeps a directory of transport user and transport provider combinations as registered by MPTN nodes. This would be appropriate for a NetBIOS user running over an SNA provider, for example.

Product Announcements and Directions

The MPTN architecture can be used for several types of applications and APIs, including CPI-C/APPC, sockets, RPC, Telnet and other TCP/IP-related interfaces, FTAM and other OSI-related interfaces, message queuing interfaces, NetBEUI for NetBIOS, and NetWare interfaces intended for Novell IPX/SPX. These could then be transported with MPTN over TCP/IP, NetBIOS, IPX/SPX, OSI transport, or SNA. The SNA network can be either APPN or subarea SNA as long as it supports LU 6.2 sessions.

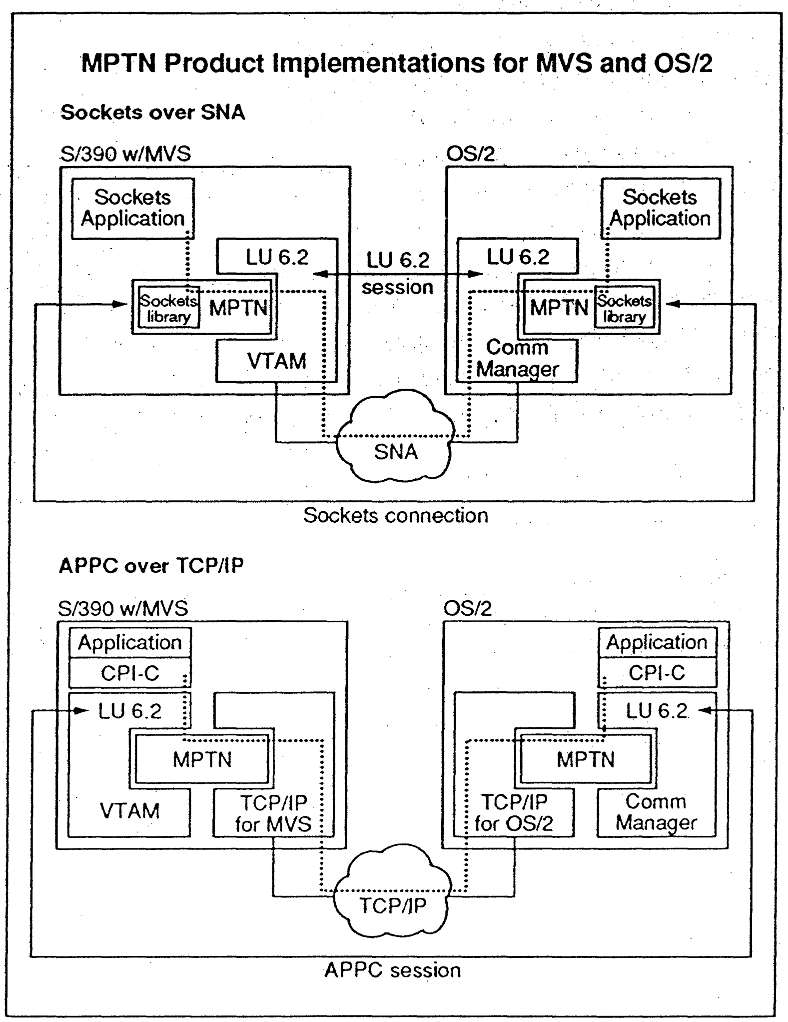

Figure 6

Although the architecture covers a broad range, actual products for each pair will be developed separately. IBM is also encouraging other vendors to use MPTN to develop other implementations. For example, at the March announcement, ki Research of Columbia, Maryland, stated that it was developing an MPTN product for DECnet to run over SNA. Several other companies are in discussion with IBM about MPTN including Hewlett-Packard, Apple Computer, and Oracle.

Product Announcements

In March 1992, IBM discussed two MPTN implementations under way—sockets over SNA (which IBM informally called SNAckets prior to the announcement) and APPC over TCP/IP. Both these sets, which are shown in Figure 6, were formally announced in March 1993 for both MVS and OS/2 platforms. General availability is scheduled for June 1993. The product name for both the MVS and OS/2 implementations is the Multi-Protocol Transport Feature (MPTF).

In theory, MPTN allows a device to have only one protocol stack installed. For the sockets-over-SNA implementation, this is the case. The MPTF includes a sockets library so the systems do not need a TCP/IP stack.

However, the SNA-over-TCP/IP implementation requires both full protocol stacks on each node as shown in Figure 6 (see page 7). For example, an OS/2 platform would need both TCP/IP for OS/2 and Communications Manager as well as the MPTF software. Only the LU and not the PU of the Communications Manager is used by the MPTF, which may reduce overhead and memory usage somewhat. But it seemed more cost effective not to create a separate SNA communication product just for use with MPTN. Similarly for MVS, the host needs both VTAM and TCP/IP for MVS to use SNA-over-TCP/IP with MPTF.

APPC over TCP/IP

The APPC over TCP/IP product will support any application with an APPC or LU 6.2 interface on MVS or OS/2. This includes CPI-C applications. It also includes other applications which usually run over dependent LU sessions, but only if these applications support LU 6.2 sessions. For example, some users with CICS, IMS, or DB2 installations have added an APPC or CPI-C interface to these applications. Support for these applications with dependent LU sessions is in IBM's statement of direction as discussed below.

Sockets over SNA

The sockets over SNA products do not support all applications which use TCP/IP. Therefore, the user cannot yet consider these products as a general solution for this environment. These products support a specific subset of application interfaces which varies with each implementation.

For example, IBM's OS/2 version of TCP/IP is based on Berkeley BSD and many applications are written to the sockets interface. Therefore, MPTF on OS/2 supports NFS, X-Window, PING, FTP, Telnet, REXEC, RSH, SMTP, RPC and Talk. However, only a sample application is provided for the domain name server.

On the other hand, IBM's TCP/IP for MVS uses a non-sockets interface for many applications. This non-sockets interface needs separate MPTN support. The current MPTF for MVS supports the following sockets applications: NFS, X-Window, and PING.

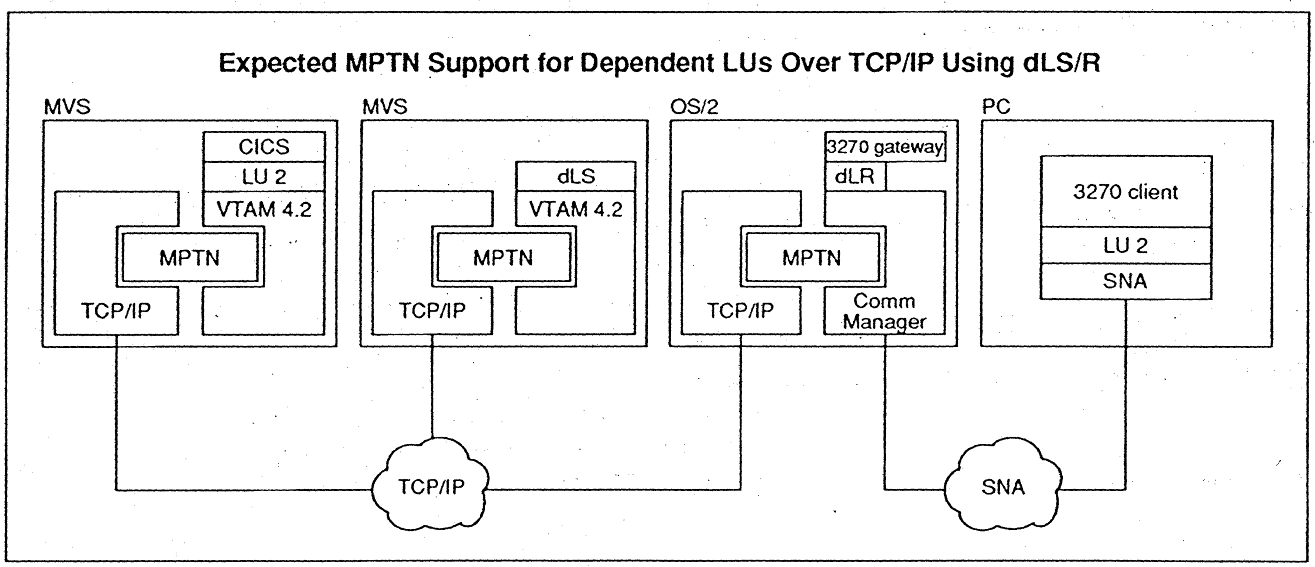

Figure 7

Directions

At the same March announcement, IBM made three statements of direction for MPTN products. *SNA Perspective* expects the products discussed in these direction statements will be available within a year.

First, IBM will implement sockets over SNA for the AS/400. Since the AS/400 is in a different line of business than Networking Systems and therefore may not be as invested in MPTN, we expect this product may take longer than the others.

Second, dependent LUs will be supported over TCP/IP. *SNA Perspective* believes that this implementation will use the dependent LU server/requester (dLS/R) model that IBM discussed as a statement of direction in September 1992. This model was discussed in detail in the November 1992 *SNA Perspective* article "Old Apps, New Nets: Supporting Dependent LUs Across APPN."

Figure 7 (see page 8) shows *SNA Perspective*'s estimate of how a dependent LU session could be implemented over TCP/IP without any SNA flows by using MPTN and dLS/R. The only SNA traffic would be between the 3270 client and the OS/2 MPTN node (and with some gateway products, even that flow is over NetBIOS or some other protocol).

Third, IBM stated that it would implement MPTN as a gateway on an OS/2 platform between SNA and TCP/IP. This will allow, for example, sockets applications in systems on a TCP/IP network to access sockets applications on an SNA-attached end system. In this case, MPTN would be required on the SNA end system and in the gateway, but not on the non-SNA end system.

Expectations

These products and statements of direction do not reflect all the MPTN products expected in the near term. However, IBM's strict rules for formal statements of direction do not allow them to discuss other expected product directions. *SNA Perspective* expects that the most likely products from IBM would be support for:

- Both APPC over TCP/IP and sockets over SNA for AIX on the RS/6000 and perhaps also on the PS/2
- APPC over TCP/IP on OS/400
- Both APPC over TCP/IP and sockets over SNA for DOS and Windows (or at least Windows)
- OS/2 gateway for APPC over TCP/IP
- NetWare IPX interfaces over APPC and TCP/IP
- LU 0, 1, 2, and 3 support over TCP/IP on DOS and Windows (or at least Windows)

MTPN and Other Approaches

As mentioned above, the function MPTN provides is far from a new concept. Multiprotocol support has always been of interest and value and the need for it is increasing every year. However, since there are many different environments, there are several approaches to the problem. One company may select more than one solution to address specific situations. Some categories of solutions are listed below:

- Choose a single protocol and change all existing installations to use the one protocol, at least on the backbone. The most popular effort toward this end is the Open Systems Interconnection. Several companies have attempted to use SNA or TCP/IP exclusively. Problems occur with this approach, for example, when departments need individual solutions unavailable on the strategic protocol or when the effort and expense of migration is very high.
- Bridging multiple protocols over a shared LAN or WAN. The most common example is the ability of SNA and TCP/IP traffic to run over IEEE 802 LANs. One drawback is that bridging has less flexibility and scalablity than protocol routing.

- Encapsulation/tunneling. For many years, a wide variety of products have allowed one protocol to be enveloped inside another for transit across the other's environment. Many multiprotocol routers today, for example, have the ability to encapsulate several types of traffic, including SNA, across IP networks.
- One-to-one conversion solutions support a specific environment or set of environments. There are many very different types of one-to-one solutions. IBM's CICS-sockets interface is a one-to-one solution for one application subsystem to run over one transport. Application-level gateways may provide conversion all the way up the protocol stack. A TCP/IP-SNA layer-four gateway is another type of solution.
- General solution. This solution is a structure for running several upper-layer types over several lower-layer types using a common set of procedures.

General Solution

MPTN is an example of a general solution to multiprotocol transport. Most of the leading system and networking companies have attempted to develop a general approach and have given up on the effort before announcing it. Several standards bodies have or had efforts in this direction such as X/Open's XTI.

Solutions for SNA over TCP/IP

Several solutions exist today or are being developed for supporting SNA applications over TCP/IP. A selection of them are shown in Table 2.

Solutions for SNA over TCP/IP

Solution	Platforms	General/ Specific	Encapsulation/ Conversion/ Other	Layer/ Level	End System/ Gateway
MPTN[1]	MVS, OS/2	General	Between conversion and encapsulation	Above transport	Either
tn3270[2]	OS/2, many	Very specific	Conversion	Session and presentation	End system
CICS-sockets interface[3]	MVS, OS/2	Very specific	Conversion	Between session and presentation	End system
APPI[4]	routers	Specific	Conversion (control) Encapsulation (data)	Above transport	Gateway
APPN over sockets[5]	6611, IETF	Specific	Encapsulation	Above transport	Gateway
APPN over IP with DLSw[6]	6611, IETF	Specific	Encapsulation	Network	Gateway
A-NET (Tuebner)[7]	MVS	Specific	Conversion	Presentation	Gateway

1 End system or gateway has MPTN base plus modules as needed for each user/provider pair.

2 3270 client on TCP/IP network has tn3270 client which intercepts 3270 datastream and sends it over telnet connection rather than LU 2. Host has tn3270 server with dummy LU 2 to access the unchanged 3270 application.

3 Sockets interface on CICS subsystem allows file transfers but not interactive traffic from sockets applications on TCP/IP. (IBM TCP/IP also needed on mainframe with CICS.)

4 Traffic between APPN ENs or LEN nodes through routers with APPI open network node. APPN control information is converted to TCP/IP directory/topology format. APPN EN-to-EN LU-LU data is sent encapsulated across IP network.

5 From APPN EN through router with both NN and TCP/IP stack crossing IP network through sockets connection to similar router to target EN.

6 From APPN EN through router with NN and data link switching (DLSw) routed across IP WAN (with no APPN nodes) to router with NN and DLSw to target EN.

7 From any 3270 workstation through 3270 host thru A-NET as telnet terminal to any Unix application on TCP/IP.

Table 2

Solutions for TCP/IP over SNA

There are fewer solutions today for TCP/IP over SNA than for SNA over TCP/IP. Although SNA is one of the most widely used protocols, both the hierarchical nature of subarea SNA and IBM's competitive posture have made multiprotocol support across SNA less common. Further, some products that were developed were too complex or had unacceptable performance.

IBM offers several options on its 3745 communication controllers to allow TCP/IP traffic to travel encapsulated over a subarea SNA network. One example is SNALink. Several other companies offer products for access from TCP/IP into SNA, including gateways from companies like Open Connect Systems (formerly Mitek) of Carrolton, Texas. (For more detailed information on these solutions, see three-part series on Supporting TCP/IP over SNA in *SNA Perspective* May, June, and July 1992.) In addition, several specific home-grown solutions are found in places where there was a significant business need.

Since full APPN support is still relatively new on platforms other than the AS/400, it is not surprising that there is little in the way of products to support other protocols across it. However, now that APPN is available or imminently available on a wide variety of platforms including the mainframe and since the APPN architecture is more amenable to multiprotocol interaction than subarea SNA, interest is rising quickly in multiprotocol solutions for APPN.

Issues with MPTN

Interested but Skeptical

The vendor and user reaction to IBM's networking blueprint and in the concept of MPTN has been generally favorable. It is congruent with the current industry belief that there will not be one API, interface, or transport, though companies are limiting them to a small set of choices. It also follows the industry desire for increased flexibility in mixing and matching pieces between stacks rather than each application being tied to a specific full stack, which IBM terms application independence.

However, these same observers are cautious regarding MPTN, wondering how MPTN can succeed when many others have tried to develop such an environment before and failed. Several papers have been published over the years about similar solutions which have not been implemented. IBM thus faces some skepticism in promoting MPTN.

Solution for Nonexistent Problem?

Another concern about MPTN is that "it is a solution for a problem that does not exist," since there are few sockets applications running on systems attached only to SNA networks and few CPI-C applications running only on TCP/IP networks. This has some validity, though there are several redundant links and redundant protocol stacks installed that could be eliminated with MPTN. In addition, MPTN could be considered an enabler for companies to use their networks in ways they were not able to before and to select applications with less reliance on the underlying transport.

The Proprietary versus Open Debate

IBM has provided significant technical information to X/Open for its consideration of MPTN. However, the company is not releasing complete technical specifications on MPTN until and unless it is accepted as a standard. X/Open seems to believe it has sufficient information on which to base its decision, however.

IBM is also promoting MPTN to other vendors and will license the full technical specifications to each vendor for a fee or technology exchange agreement. Some would argue that IBM should just throw MPTN into the public domain. However, since IBM is offering the opportunity to the industry to accept MPTN as a standard and will provide it at no charge if it becomes a standard, it seems fair to us, if it is not accepted, that IBM use MPTN as any other intellectual property.

Moves Rather than Solves Problem

Some have noted that MPTN does not eliminate the problems of multiprotocol transport but rather raises the complexity to another level. This is a valid statement, but would be true of any solution short of eliminating multiple protocols. IBM chose to put its common transport semantics above the transport because it was the least complex and most beneficial level for several reasons which are described in more detail in the reference documents listed at the end of this article.

May Still Need Both Stacks

Although MPTN is designed to reduce redundant protocol stacks, some implementations will use multiple stacks for practical reasons. For example, for SNA-over-TCP/IP, IBM chose to use the LU 6.2 capabilities of the existing OS/2 Communications Manager product rather than extract the LU components to use with MPTN. This may turn out in some cases to be more expensive and require more resources on the system, but in other cases could actually be less expensive and easier to develop by taking advantage of existing products.

IBM Agenda—Sell APPN

Some argue that MPTN is part of IBM's agenda to increase the popularity of APPN. MPTN certainly makes APPN easier to use under the popular TCP/IP-related suite of applications. Certainly, IBM would like to position APPN as a better technical solution than TCP/IP for a single-protocol backbone or as the primary protocol in a multiprotocol environment.

On the other hand, MPTN also makes TCP/IP easier to use under popular SNA applications. IBM product introductions to date indicate that the company is using MPTN to target both markets.

XTI Feedback

Although MPTN and IBM's other proposed XTI extensions are still wending their way through the recommendation process in X/Open, X/Open's response can be characterized in three short phrases: it loves the SNA interface, likes the transport compensation, and is not much interested in the gateway. The addition of SNA to the XTI interface is practically assured. X/Open is also actively considering the proposal for transport compensation. However, X/Open seems to believe that XTI and/or MPTN in an access node is more appropriate to standardize than the gateway configuration. IBM is likely to win two out of three in this contest.

What About Middleware?

Since middleware is a relatively new concept, some users might wonder whether MPTN is another type of middleware. There are certain similarities but there is an important difference. Middleware is intended as a common API for all application types while MPTN is designed to support different applications and APIs with only minimal changes. Instead of developing a set of transport interfaces, middleware vendors could use MPTN—providing a common API with middleware and using MPTN to provide a common interface to protocols from the middleware.

Need To Change Existing Software

In order to have the application be oblivious to the transport selection, MPTN will usually require a small change to the application interface. For example, most sockets interfaces would need to be adapted to point to the user request function (allowing other protocols to then be selected) instead of binding to TCP/IP at the time of socket creation.

IBM says that MPTN products will include software that automatically replaces the appropriate component in the interface.

Need MPTN Everywhere

Some users are concerned that MPTN is required at both ends of any multiprotocol connection or at least at two points between the two applications, such as on gateways. It does not interact at this point with other solutions such as RFC 1006 and these other products cannot interpret the MPTN headers. This is not a specific problem with MPTN but would be an issue with any similar solution, an issue which users must keep in mind.

Support Needed for Dependent LUs

For SNA applications, MPTN currently does not support sessions with any of the dependent LUs. It supports mapping only from applications with an LU 6.2 or Advanced Program-to-Program Communication (APPC) interface. (An APPC interface has been available for some years for the primary traditional host-based dependent LU applications such as CICS, IMS, and DB2.) IBM has made a statement of direction that it will support dependent LU applications in MPTN in the future. *SNA Perspective* expects that support will be provided in conjunction with IBM's Dependent LU Server/Requester model which we believe will be available sometime in 1994.

Conclusions

SNA Perspective considers MPTN in concept and design to be beneficial to users. The need for such a product is increasing. However, IBM needs to roll it out quickly and effectively.

The company must maintain an aggressive release schedule for new MPTN products. To be accepted as a simple, cost-effective solution, additional MPTN implementations must be perceived to be quick and easy to develop. Therefore, a usable set of implementations must be available in a short time frame and additional products should follow close behind. If IBM cannot bring these products to market quickly and easily, the image of MPTN will be tarnished and the opportunity may be lost.

In order for MPTN to be successful, IBM must succeed either in having it adopted as a standard or recommendation and/or in gaining cooperation from several other vendors in creating implementations.

MPTN's success would certainly further the cause of APPN since it would allow APPN networks to be more easily interfaced with TCP/IP networks. At the same time, however, MPTN eases SNA applications running over TCP/IP and so does not seem to us to lock users into one protocol environment or the other.

References

MultiProtocol Transport Networking: Formats, GC31-7074-00, IBM Corporation, 1993.

Multiprotocol Transport Networking: Technical Overview, GC31-7073-00, IBM Corporation, 1993, 59 pages.

Dynamic Protocol Selection for Socket Applications, David Ogle, et. al., IEEE Networking, June 1993.

"Communications for Cooperating Systems: OSI, SNA, and TCP/IP", R. J. Cypser, Addison-Wesley, Reading, MA, 1991, 743 pages. ■

Architect's Corner

Mixing Oil and Water

by Dr. John R. Pickens

SNA and TCP. 3270 and LU 6.2. Sockets and CPI-C. Oil and water. Much interest is being expressed these days in mechanisms for intermixing previously independent protocol environments.

One user wants to build an SNA backbone but run applications built originally for the TCP/IP framework. Another user wants to build an IP backbone, but run applications built originally for the SNA framework. A third user wants to run 3270 on the end systems but over an APPN backbone. Oil and water.

How can they be made to mix? How well do they mix? Certain combinations mix poorly. For example, a desire to interwork between applications written to CPI-C (usually SNA) and sockets (TCP/IP) will disappoint and frustrate. Likewise doomed to failure is a desire to have 3270 terminals access X-Windows clients.

The universal panacea for mixed protocol environments, the transparent any-to-any protocol converter continues to elude discovery—the anti-graviton of the protocol world.

tn3270

Other combinations, however, seem promising. The 3270 datastream, whose most natural form is 3270 over LU 2 (3270/LU 2), can be adapted to other forms. RFC 1041 defines a mechanism for running 3270 datastream atop telnet/TCP. Telnet negotiation is used to define the terminal type. The 3270 datastream is then encapsulated, but otherwise unmodified, in the telnet/TCP packetized envelope.

It is not perfect—not all characteristics of a 3270 terminal are supported, such as the ability to toggle between the LU-LU and SSCP-LU sessions. But it works. It may even work well enough for most environments.

appc3270

Another combination, 3270/LU 6.2 is also possible. Using a technique similar to the above RFC, the IBM Market Enablement Group, with lead designer Tim Huntley, is currently defining such a mapping—carrying the negotiation in APPC GDS variables, rather than telnet negotiation options. Similar functionality. Similar restrictions. Will it work? I think so. Is it transparent? Mostly. But, like tn3270, changes are required both in the end system "client" and in the mainframe "server."

MPTN

Continuing the focus on end systems, one further combination seems to be promising—IBM's proposal for Multi-Protocol Transport Networking (MPTN). The concept of MPTN is simple, though difficult to get right—enable within the end system the ability for specific service interfaces, e.g. CPI-C and sockets, to be flexibly layered atop multiple middle-layer transport stacks. For example, CPI-C (an SNA-style interface) running atop TCP/IP transport. Or, sockets (a TCP/IP-style interface) running atop APPC transport. But the intermixing stops at the middle layers—CPI-C applications still talk to CPI-C applications; sockets applications still talk to sockets applications.

The key to MPTN is the definition of additional elements in the protocol datastream, called compensations, which add missing functions end to end. So, in the CPI-C case, the underlying protocol becomes CPI-C–compensations–for–TCP. For sockets, the underlying protocol becomes sockets–compensations–for–APPC. (See the article in this issue and its listed references for more details.)

Will MPTN work? I think so. What are the risks? Only one in my opinion—widespread adoption. But the technology works.

appc3270 over TCP

Before leaving end systems, one additional SNA/TCP combination is possible—a variant of appc3270. Using MPTN concepts, appc3270 could be layered atop TCP transport (just like CPI-C/TCP). Then tn3270 would become obsolete. Who needs telnet anyway?

APPN over TCP/IP

Moving one level deeper into the SNA/TCP internetworking, additional levels of mixing are also possible. At the routing layer, TCP/IP can be defined as a datalink between routers—or as IBM calls it, a shared transport access facility. IBM has recently defined just this mapping for use by APPN/ISR routing and made it available on the Internet as IETF draft-kushi-appn-tcpip-00.txt. This mapping was first demonstrated by IBM at Fall 1992 Interop in San Francisco.

I am sure that APPN/HPR, sometimes called APPN+, will define another IP-compatible datalink mapping, even more efficient than APPN/ISR.

LLC2 Tunneling with DLS

At layer two (bridging extensions), one additional form of mixing is possible—LLC2 tunneling. IBM has a protocol for layer-2 tunneling called Data Link Switching (DLS). Documented in informational RFC1434, DLS defines (a) locally terminated LLC2 connections (also capable of supporting SDLC and NetBEUI), (b) a discovery function for dynamically locating MAC addresses within the network of DLS capable routers, and (c) a packet forwarding function (with flow control) for forwarding packets using TCP across IP backbones.

Does DLS work? Yes. What are it's risks? First, vendor adoption. Presently at layer two, each vendor has defined its own protocol. The APPI consortium is proposing yet another tunneling standard (possibly incorporating a tunneling protocol from McDATA).

The second risk is that it is difficult to fully support the bridging functions at layer 2. DLS, for example, requires an extension to support non-source routed environments (e.g., Ethernet). With source routing, a phantom virtual ring is defined inside the internetwork so that looping discovery frames (incoming to the phantom virtual ring) can be detected and killed. No such construct exists for non-source routed environments. While I do believe that such a function can and will be defined for DLS (and its cousins from other vendors), I believe that routing (a.k.a. APPN), which suffers no such deficiencies, will become the preferred solution.

So Many Options

In protocols, it seems that mixing of oil and water is here to stay. Unlike the metaphor, protocol oil and water can be mixed in more than one way—at the end system, at the intermediate system for routing, and at the intermediate system for (extended) bridging.

A personal confession—I experience a certain level of discomfort at the complexity that has been introduced into internetworking. Others tell me I am not alone. For both vendors and users, the myriad protocol-mixing combinations that are exploding onto the internetworking landscape are unsettling. The support burden is scary. As protocols evolve, can the mixtures also evolve at the same rate? How vigorously must we shake the internetworking container to keep the protocols well mixed? What price are we paying for our inability to standardize on a single protocol suite? Is this protocol mixing even relevant as we migrate to gigabit protocols?

But perhaps I am overreacting. While the world has not become simpler, at least we now have options. ■

• Subscription rates: U.S. - one year $350, two years $520. International - one year $385, two years $590 • *SNA Perspective* is published monthly by CSI, 2071 Hamilton Avenue, San Jose, CA 95125 • Telephone (408) 371-5790 • Fax (408) 371-5779 • Managing Editor: Louise Herndon Wells • Associate Editors: Vincent Busam, Basil Treppa • Circulation Coordinator: Cheryl Roberts • Contributors: Dr. John R. Pickens; Diane Pozefsky, IBM Corporation • Typesetting and illustration: Aaron Lyon at dSIGN