Volume 13, Number 8
August, 1992
ISSN 0270-7284

SNA Perspective

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

Blueprint to Integrate the Architectures

IBM's Networking Blueprint was introduced in March 1992 as a strategy to support the coexistence of a wide range of application interfaces over a diverse set of networking architectures through a common set of transport semantics. For example, the Networking Blueprint is intended to support connections between applications using the same local interface (CPI-C, RPC, MQI, X.400, FTAM, etc.) over dissimilar transport networks. The IBM Networking Blueprint strives to:

- Integrate diverse LAN, WAN, and transaction processing technologies
- Support multiprotocol, multivendor, and multimedia elements
- Enable users and their applications to interconnect across diverse environments
- Support client/server computing in a consistent way for the end user
- Exploit high-speed, high-bandwidth technology and services
- Provide comprehensive, architecture-independent management

The promise is to reduce duplicate resources while preserving end-user applications. After reading this article, the reader will have insight into the IBM Networking Blueprint objectives, the means whereby it proposes to provide for interoperation of diverse applications over multiprotocol environments, particularly at the transport level, and the IBM products that are emerging to implement the Networking Blueprint.

(continued on page 2)

Data Link Switching on the IBM 6611

The IBM 6611 Network Processor is scheduled for general availability in September. This product is IBM's long-awaited entry into the hot world of multiprotocol internetworking. As might be expected, the 6611 has a number of features that are targeted at the world of existing SNA networks. These features and some others have been placed under an umbrella term called data link switching.

This article focuses on data link switching, rather than reviewing the entire 6611. It explains what data link switching is, how it got its name, what internetworking problems it was designed to solve, and its features. It also compares several of these features with similar features in products from other vendors.

(continued on page 10)

In this issue:

Blueprint to Integrate the Architectures......1
IBM drew up the Networking Blueprint to address user questions about interoperability and the future direction of IBM data communications. The Blueprint is comprehensive and far-reaching, and is unprecedented from IBM. This article describes the Blueprint in detail—the components that exist today, how it fits with X/Open XTI and IEEE POSIX, what users can plan to build from it, and whether common transport semantics is a vision or a dream.

Data Link Switching on the 6611................1
DLS uses neither bridging nor routing but something in between to support SNA and NetBIOS over router networks. Learn four main DLS tools: local termination, link conversion, route caching, and congestion control. Two DLS benefits are reducing the impact of chatty LAN protocols on WANs and supporting older non-LANable SDLC devices.

Architect's Corner: Protocol Engineering —Three Futures.....17
Aren't today's protocols good enough? How should they change? Our architect examines three camps—the universalists, the fundamentalists, and the revolutionaries.

(continued from page 1)

End-user applications and the network infrastructures that provide for their connectivity have become increasingly complex. Historically, environments were often characterized as either SNA or non-SNA. This relatively simple distinction has evolved into production networks that use a variety of interfaces (APPC API, CPI-C, MQI, RPC, and NetBIOS) connected through a range of transport services (SNA subarea, APPN, TCP/IP, connectionless or connection-oriented OSI transport, and IPX) over a collection of possible WAN and LAN protocols (SDLC, HDLC, token ring, 802.3/Ethernet, and X.25).

Unfortunately, most users are unable and/or unwilling to retrofit to a single-approach solution. The business reality is that incompatible networks have evolved and include multiple, department-level client/server solutions. Users need these solutions to coexist in a cost-effective way.

That is, users want to move from multiple, separate network infrastructures to support a heterogeneous mixture of applications, interfaces, transport protocols, and underlying subnetwork links and adapters. A solution that can cost-effectively eliminate networking redundancies would be heralded as a remedy for customers beset by the costs and complexities of multivendor, multiprotocol, and multimedia networked environments.

A second and interrelated challenge is to provide for interoperability among a wide range of hardware and software so that client/server applications that are distinct in geography, processor platforms, and runtime environments can cooperate meaningfully regardless of distinctions in native session services, transport protocols, or preferred subnetwork interfaces.

Enterprise Networking Requirements

Users and their applications must interconnect and share resources in increasingly complex and heterogeneous networked environments. It is not uncommon today for user networks to include SNA, OSI, TCP/IP, NetBIOS, IPX, DECnet, Appletalk, and others. Valiant attempts by standards organizations to standardize-by-committee have not yet provided "standard standards." Many router and bridge vendors have entered this marketplace in recent years to address these incompatibilities and have themselves further complicated matters through vendor-specific, incompatible solutions.

Unfortunately, end users and their applications need to interconnect and share resources, but cannot generally do so between and among dissimilar networked approaches. A costly corollary has been the development of multiple and redundant network infrastructures in an organization where, for example, an SNA network transports SNA data, a TCP/IP network transports TCP/IP data, and an OSI network transports OSI data.

Significant user requirements in these increasingly unnavigable environments include the following:

- Write only once an application that runs over different protocols
- Support applications based on the functions they provide, and not on the underlying networking protocols they use
- Allow these applications to run anywhere on the internetworked environment
- Enable applications to send and receive standard calls to and from the network interface while the selected network infrastructure is kept transparent to the application, end user, and to the application developer
- Provide reliable, architecture-based solutions to all problems of incompatibility, which promotes resistance to platform and operating environment obsolescence
- Develop and manage heterogeneous networks as a single network
- Consolidate multiple and heterogeneous network protocols and traffic over common adapters, thereby enabling downsizing of redundant network resources
- Provide common transport end-to-end across heterogeneous subnetworks

Networking Blueprint Objectives

The objectives of the IBM Networking Blueprint respond to the requirements described above and address emerging networking and networked application technologies. These major objectives are:

- Integrate LAN, WAN, and transaction processing technologies to enable applications running in diverse processing platforms and operating environments to interconnect and share resources coherently, regardless of underlying differences in application session services, network transport and routing algorithms, and network connectivity interfaces
- Support multiprotocol, multivendor, and multimedia networked application elements in a consistent way to enable users to select from a range of products running across a variety of platforms
- Support client/server computing among these diverse environments in a consistent way to the end user
- Exploit emerging high-speed, high-bandwidth transmission technology and carrier services including, for example, standards-based carrier services—such as asynchronous transfer mode (ATM)—as well as the extension of current network protocols to support variable length packets, diverse priority algorithms, spectrum-shaping of offered loads, bandwidth utilization metrics and optimal route assignment
- Provide comprehensive, architecture-independent systems and network management in distributed, centralized, and hybrid environments

Expressing the Blueprint

Clearly, developing products that support such complex user requirements in a consistent way is an enormous task. As of this writing, IBM has formally only announced Advanced Peer-to-Peer Networking (APPN; see *SNA Perspective*, April 1992) as a transport implementation of the Networking Blueprint, although it has discussed plans for others, as described below.

By definition, of course, IBM products that implement common programming interface for communications (CPI-C) over logical unit 6.2 (LU 6.2) and also select APPN for transport express the Blueprint.

Elegant Model

As described below, *SNA Perspective* believes that IBM has articulated the Networking Blueprint transport architecture in an elegant way. It remains to be seen, however, to what extent upcoming products express this set of architectures coherently. It seems likely to *SNA Perspective* that IBM's annual September announcement avalanche may be a likely time for the company to begin delivering on the Networking Blueprint vision presented in March.

The key challenge for IBM will be to express the Blueprint in products in such a predictable way that customers feel that they can count on it, that they can expect future product introductions to be compatible with present ones. This perception of resistance to obsolescence is key to acceptance and requires that the underlying architectures be stable.

Networking Blueprint Description

Figure 1 (see page 4) presents an overview of the IBM Networking Blueprint. The intrinsic power of the Blueprint lies in its stated objective to enable applications that support consistent interfaces (e.g., CPI-C-to-CPI-C, RPC-to-RPC, MQI-to-MQI) to interconnect and share data resources regardless of the underlying session services, transport protocols, or network interfaces supported by each participating application.

Application platforms expressing the Blueprint would:

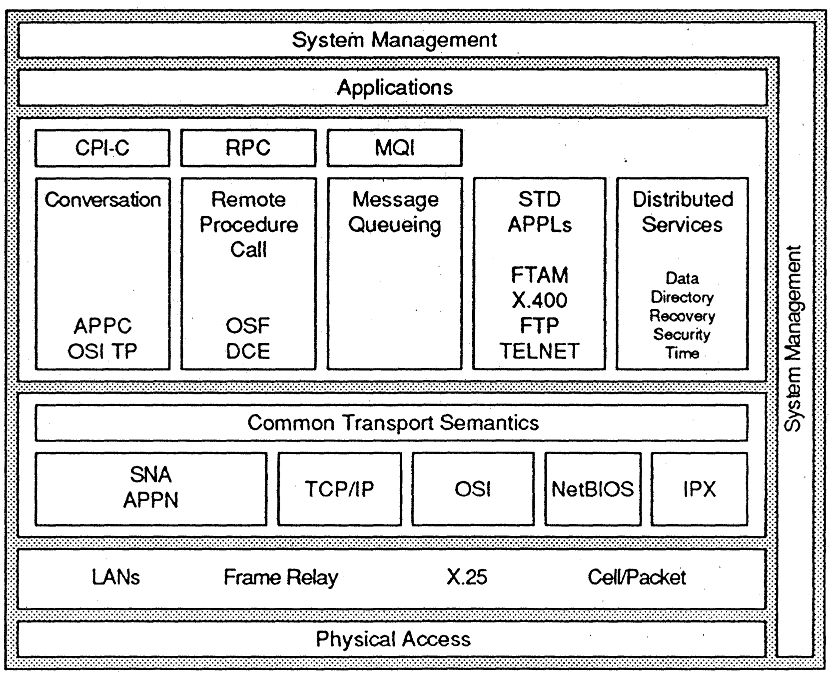
- Empower end users to focus on the task at hand by providing transparent access to target (server) applications and data without being encumbered by multiple interfaces
- Fulfill management's goal of reducing costs associated with multiple, overlapping network interfaces and delivery infrastructures

CPI-C, RPC, and MQI

The Networking Blueprint supports several major application services with respect to the client/server model:

- Conversational, which is supported by CPI-C access into advanced program-to-program communications (APPC) and Open Systems Interconnection (OSI) transaction processing (TP). APPC and TP applications trace their ancestry from host-centric networks and have evolved toward LAN-based client/server computing. In a conversation, each end performs explicit sends and receives and each end must remain aware of the state of the remote program.
- Remote Procedure Call (RPC), in which a local program or user calls a procedure (subroutine) that is outside the calling program. Upon execution of the remote procedure, control is returned along with any data and return code to the following sequential instruction in the calling program.
- Message Queueing Interface (MQI), in which messages are sent to and received from local/remote programs. In essence, MQI is a message delivery interface that integrates message delivery, data translation, security, queueing, naming and error recovery in a user interface that supports messaging through the use of verbs.

IBM Networking Blueprint



CPI-C = Common Programming Interface-Communications
LU 6.2 = Logical Unit Type 6.2
OSI = Open Systems Interconnection
TP = Transaction Processing
RPC = Remote Procedure Call
OSF = Open Software Foundation
DCE = Distributed Computing Environment
MQI = Message Queueing Interface
FTAM = File Transfer Access & Management
FTP = File Transfer Protocol
SNA = Systems Network Architecture
APPN = Advanced Peer-to-Peer Networking
TCP/IP = Transmission Control Protocol/Internet Protocol

Figure 1

The key Networking Blueprint promise is to connect applications that each use the same interface (CPI-C, RPC, MQI, FTAM, X.400, FTP, Telnet, etc.) through various possible transport protocols (APPN, TCP sockets, OSI transport, NetBIOS, IPX) over a heterogeneous range of possible LAN and WAN services. The immediate result is that freedom of choice will be provided at any layer without requiring a specific selection or particular implementation of function at another layer.

That is, applications and their support services will be selectable as independent of underlying session, transport, network, and data link protocols. Further, layer-specific choices will not be negatively impacted as technology evolves in other layers. Also, a range of choices could be made in a given layer without impacting choices made at other layers.

No Management Specs Yet

Figure 1 also indicates that the Networking Blueprint defines systems management functionality throughout all of the layered environments. *SNA Perspective* believes that IBM will, over time, develop distributed as well as centralized network and systems management tools to provide for this capability, likely to be based on SystemView. As of this writing, IBM has not released Networking Blueprint systems or network management specifications.

Problems with Multiple Transports

One of the most significant promises of the Networking Blueprint is the elimination of the problems generated by multiple transports or multiple protocol stacks in each end system. There are several problems associated with colocation of multiple, complete protocol stacks within the same processing environment, especially in workstations.

- Multiple active protocols. This can create buffer conflicts, contention for limited address space and range, and limitations on numbers of available links as provided by adapter cards. Further, several links are not multiprotocol-enabled.
- Duplicate infrastructures. This is the major problem facing users today, as has been stated earlier. The coexistence of multiple protocols in the enterprise has given rise to duplicate network infrastructures with duplications in costs for adapters, interfaces, lines, network processors, and staff.
- Transport-bound applications. Not all workstations, midrange processors, and hosts could possibly implement all possible desired protocol stacks. Therefore, opportunity costs arise for a large segment of the user population.
- Code management. The design and coding effort needed to maintain and coherently evolve duplicate infrastructures for networked applications is completely out of proportion to the return. Environments may include up to dozens of protocols, each with intrafamily variations in functional version and release levels.
- Memory cycles. Coresidence of multiple protocols stacks requires significant additional memory and cycle usage, and also significantly increases the cost burden associated with multiple software licenses, code libraries, disk space, and the global expenses associated with version/release upgrades.
- Management. System and network management approaches vary considerably among various protocol stacks and tend to conflict with each other, especially under stressful conditions. This results in, at minimum, unacceptable recovery periods.

Common Transport Semantics

The Networking Blueprint addresses the above-stated problems of multiple transports through the use of common transport semantics. That is, the Blueprint is intended to enable the running of many types of applications and their services over several major networking protocols. For example, APPC applications could run over OSI or TCP/IP and applications written to the TCP/IP sockets interface could operate over SNA. These capabilities are engendered by making the multiprotocol transport network layer opaque from the layers above it through common transport semantics.

Multiprotocol Transport Network

The Networking Blueprint concept of common transport semantics is currently most clearly presented by IBM's multiprotocol transport network (MPTN). An MPTN is a way to support or connect a collection of single-protocol transport networks (SPTNs), each of which has its own transport protocol. MPTN can interconnect these SPTNs through MPTN gateways or servers. In all cases, the end users and their applications will be unaware of which protocol or collection of protocols is selected to transport data across the network.

The major significance of this approach is that MPTN would be able to provide connections between applications that use the same interfaces (e.g., CPI-C, RPC, MQI, FTAM, X.400) without any resulting networking protocol dependencies. For example, today applications written to NetBEUI (NetBIOS End User Interface) can run over NetBIOS or IPX; applications written to CPI-C can run over APPC/SNA or OSI-TP; applications written to an appropriate subset of X/Open Transport Interface (XTI) can run over TCP/IP, OSI, or NetBIOS. However, it is not possible today for these applications and their associated interfaces to run generically over other transport networks.

An MPTN gateway would be a transport-level gateway between two or more protocols, such as SNA and TCP/IP. In this way, for example, a pair of RPC applications, one of which is located on an SNA network and the other is on a TCP/IP network, could interface with this transport-level gateway rather than requiring an application-level gateway. This MPTN gateway could be used to connect multiple SPTNs to create an integrated heterogeneous network.

Another MPTN solution would take the form of a multiprotocol server. For example, a server application on a node with MPTN could provide support for SNA, OSI, and TCP/IP and this single server would be able to support clients on SNA, OSI, and TCP/IP networks. The actual MPTN routing function would likely maintain transport addresses used in each protocol group as unique.

MPTN and the Standards Process

IBM made a proposal to X/Open this year to enhance its X/Open Transport Interface (XTI) through inclusion of several aspects of MPTN and other elements. Some of these proposed changes are more likely to be accepted than others. Before we go into detail, a little background will be helpful.

X/Open, a consortium of vendors developing specifications based on official and de facto standards, had developed XTI to define a transport service interface independent of any specific transport provider. XTI is concerned primarily with a usable interface to OSI connection-oriented and connectionless transport but has been extended to include transmission control protocol (TCP), user datagram protocol (UDP), and NetBIOS. In addition, X/Open is cooperating with the IEEE 1003.12 committee which is developing the POSIX Detailed Network Interface, essentially a standardized version of XTI.

IBM's MPTN proposal has two elements that may interest our readers, though they would both have long-term rather than short-term effects on readers' networks.

The first element proposes two extensions to XTI that advise application developers how to use XTI to access SNA and IPX. These seem to be of great interest to X/Open members and are likely to be considered quickly and probably adopted. With the SNA extension to XTI, for example, a developer could take an application written to XTI with TCP/IP and adapt it to run over SNA though a set of C library functions.

The second element is a proposal for a set of generalized compensation protocols. MPTN proposes a base set of transport services. (This base set is not just the "lowest common denominator" of services that are common to all protocols, nor is it an exhaustive set. Instead, IBM has selected those services it considers to be generally useful and necessary for transport. If an application needs a function in a transport service that is not provided in the transport that is used, a compensation is performed by MPTN.) This part of the proposal is more controversial. It would affect neither users nor application developers but, rather, the smaller community of gateway developers. These gateway vendors currently have proprietary solutions for transport gateways that are either specific (A-to-B) or generalized (any-to-any). Some X/Open members believe that, since few players would be affected by this proposal and they already have existing solutions, there may not be a clear business case for going through the effort of standardizing. See Table 1 on page 7.

Multiprotocol Transport Network: Transport Services

Transport services supported by various transport providers

Transport service	SNA	NetBIOS	TCP/UDP	OSI
Connection data and termination data	Supported for native connections; compensation required for non-native connections	Compensation required	Compensation required	Supported for native connections; compensation required for non-native connections
Datagram	Compensation required	Supported	Supported	Supported
Multicast	Compensation required	Supported	Supported, see RFC 1112	Compensation required
Expedited data	Compensation required	Compensation required	Supported via Urgent Data	Supported
Native data delivery model	Record	Record	Stream	Record
Full duplex connections	Compensation required	Supported	Supported	Supported

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

Connection-oriented transport services expected by common transport users

Transport service	SNA transport user	NetBEUI transport user	XTI TCP transport user	Sockets TCP transport user	XTI OSI transport user
Connection data	Used (BIND image including sequence number & RH)	Not used	Not used	Not used	Used by OSI session layer
Termination data	Used (UNBIND image including sequence number & RH)	Not used	Not used	Not used	Used by OSI session layer or between transport and session layers
Record or stream	Record	Record	Stream	Stream	Record
Expedited data	Not used	Not used	Used	Used	Optional
Maximum expedited data size	N/A	N/A	1	1	16
Expedited marking	N/A	N/A	Not used	Used	Not used
Close type	Duplex abortive	Duplex abortive	Simplex orderly or duplex abortive	Simplex orderly, duplex orderly, duplex abortive	Duplex abortive

Conectionless transport services expected by common transport users

Transport service	SNA transport user	NetBEUI transport user	XTI UDP transport user	Sockets UDP transport user	XTI OSI transport user
Unicast datagrams	Not used	Used	Used	Used	Used
Multicast datagrams	Not used	Used	Not used	Used	Not used

Source: IBM Corporation

Table 1

Two Applications

Two interesting mixtures based on the use of common transport semantics have been discussed by IBM:

- Sockets Over SNA (SNAckets)
- CPI-C Over TCP/IP

IBM made a statement of direction (SOD) on March 25, 1992 to provide SNAckets as a direct means of enabling RPC-interfacing applications that comply with Open Software Foundation (OSF) Distributed Computing Environment (DCE) to call a TCP sockets library in common transport semantics, gain access to LU 6.2 for session services, and access APPN (rather than IP) transport services. This relationship is illustrated in Figure 2.

The reader should note, from the figure, that implementations of the Blueprint may lead to unexpected paths through the protocol stack. The path in the figure appears to be inelegant, but makes most efficient use of LU 6.2. This is an example of how the challenge of mapping and compensation is more than an academic exercise, in this case from the TCP/IP layers to SNA's logical units, components of which are split between layers four and five. In this case, use of LU 6.2 protocols allow common transport connections to reuse existing sessions.

IBM has also recently announced that it is working with a customer to prototype CPI-C over TCP/IP. This would enable CPI-C–interfacing applications, which generally use APPN for transport, to be passed over an IP network. This relationship is shown in Figure 3.

Future Common Transport Applications

Several additional application/transport combinations will undoubtedly emerge from common transport semantics. *SNA Perspective* believes that MPTN could also assist in the acceptance of APPN as a transport mechanism.

Problems with Common Transport Semantics

In essence, common transport semantics can be considered as "glue" which allows $Layer_N$ of protocol $stack_A$ to transparently coexist with either $Layer_{N-1}$ or $Layer_{N+1}$ of protocol $stack_B$. This approach can be articulated simply enough, but the ability of IBM or any company or organization to actually accomplish it is dubious, though a success could have enormous and far-reaching positive impacts.

Although, as noted above, there are several problems with multiple transport protocols, there are also several problems with a single solution. The first problem IBM faced was developing a generic

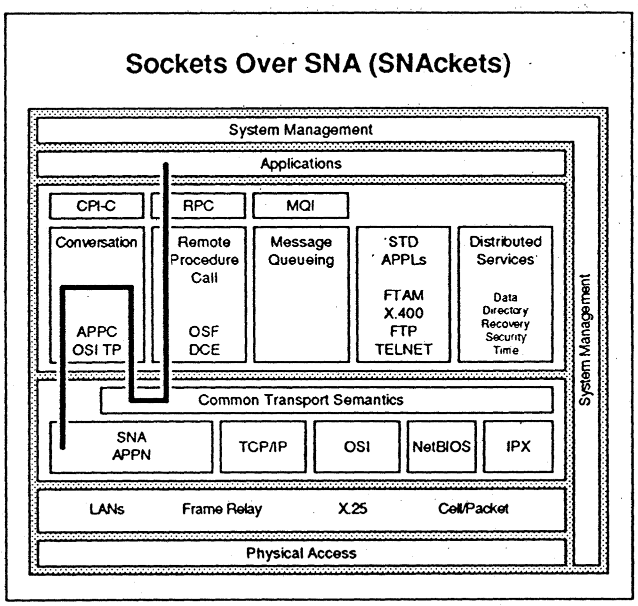


Figure 2

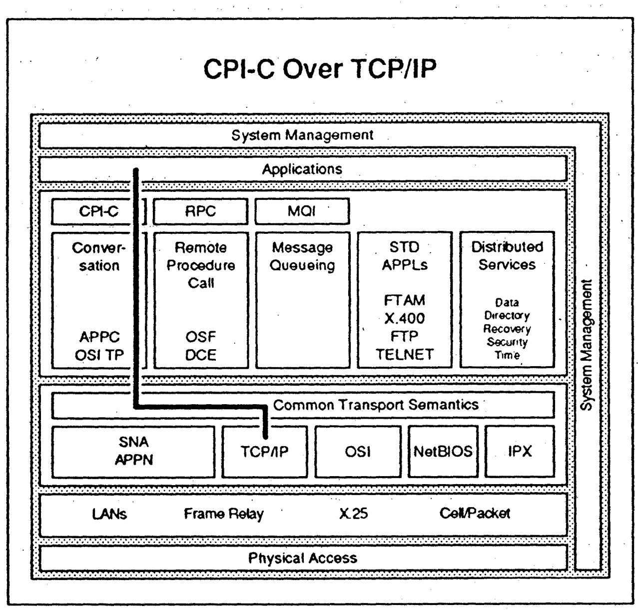


Figure 3

architecture that can provide an acceptable solution for a wide enough range of proposed uses—one that can provide a broad enough base set of services with an adequate degree of mapping and compensation at an acceptable level of ease of use and efficiency. The second problem IBM faces is convincing the vendor and user members of an appropriate forum that its solution provides what it promises.

SNA Perspective believes that IBM is likely to be successful in X/Open with the part of MPTN that extends XTI to run over SNA, but faces a greater challenge in having MPTN accepted as a generic multiprotocol transport gateway. We also believe that the concept of common transport semantics is, currently, just that—a concept. This part of the Networking Blueprint certainly sheds light on an important industry trend toward supporting applications over several transport protocols, and we also believe that IBM's MPTN work is an important contribution to that effort. However, readers should not expect that a single interface will emerge in the forseeable future to serve as a common transport layer protocol boundary in the way that the IEEE 802.2 link layer control protocol serves between the network and data link layers for LANs. Instead, readers should consider, in their long-range planning, that they will be able to maintain and integrate the major transport protocols under a variety of applications and APIs on different platforms and operating systems.

Subnetworking Solutions

One immediate and significant result of achieving multiprotocol stack integration through the use of common transport semantics would be that all supporting protocols stacks and architectures could share the same physical adapters, links, and subnetworks. From this perspective, each participating application could connect to any other such application through any of several possible WAN and/or LAN links, including SDLC, X.25, ISDN, frame relay, IEEE 802.n LANs, FDDI, ATM, SMDS, or serial optical channel (SONET).

Issues and Directions

The IBM Networking Blueprint extends great promise to users beset by complex and costly networking choices. Clearly, users will not undo or retrofit their enterprise and departmental multivendor, multiprotocol, multimedia environments. The Blueprint holds the promise of enabling the coherent coexistence of these compound solutions in a way that also reduces costs.

The question looms whether IBM will be able to deliver on the Networking Blueprint with a range of product offerings. *SNA Perspective* believes that, while the challenges and development implications are not insignificant, IBM will likely begin to deliver Blueprint-implementing products before the end of 1992. As stated earlier, APPN and its supporting products is already positioned to express the Blueprint. It seems quite likely, just from a cursory inspection of the Blueprint elements, that IBM will increasingly integrate LAN, WAN, and transaction processing product offerings throughout the SAA and AIX platforms, to Blueprint and MPTN specifications. Candidate platforms certainly also include the 6611 family.

IBM cannot afford a long absence of products based on this model, or it could suffer from market disparagement as SAA and SystemView have. Should IBM begin to deliver on this significant Networking Blueprint promise before the end of 1992, it would go a long way to creating user confidence in the existence of a stable, well-articulated architectural framework for the 1990s and beyond.

Reference

"An Introduction to Multi-protocol Transport Networking (Revised)," April 17, 1992, K. Britton et. al., IBM Corporation. ■

(continued from page 1)

Data Link Switching

Each capability in the 6611 has an equivalent in at least one of the internetworking devices from other vendors. However, the 6611 is unique in having a particular set of its features grouped under a common term, data link switching (DLS), and it implements these capabilities differently from most other products.

DLS is a technique for transporting SNA and NetBIOS traffic through a multiprotocol network in as efficient a manner as possible. One of the primary goals of DLS is to provide this transportation of SNA and NetBIOS data without affecting the end-system applications.

DLS: Between Bridging and Routing

The networking industry has converged on some common naming conventions for switching elements that operate at different levels of a multilayered communication architecture. Bridges operate exclusively at layer 2 to connect two networks together at the data link layer. Routers operate at layer 3 to connect two networks at the network layer. Both bridges and routers can be used to connect similar or dissimilar networks.

DLS, however, lies between bridging and routing. DLS terminates its data link control instances, as routers do, but does not perform routing based on a network layer header. On the other hand, while it is true that all bridges are software devices and exercise some control over the copying and forwarding of frames, DLS provides much more functionality at the data link level than bridges do. DLS does not provide routing capabilities in the 6611 for SNA as the IBM 3745 Communications Processor does. This is discussed below under No SNA Routing.

Problems Addressed by DLS

Increase WAN Efficiency of LAN Protocols

The primary problem addressed by the DLS architecture is increasing the WAN efficiency of LAN-based protocols. Most LAN protocols were designed to work on high-speed networks with minimal bounded delays. Wide area networks are relatively low speed when compared to LAN speeds and often do not provide the delay characteristics required by LAN protocols.

Accommodate Older SDLC Devices

A secondary problem that DLS addresses is a rather pragmatic one—the accommodation of older SNA devices based on synchronous data link control (SDLC) within the construct of today's internetworking environment. SDLC was designed to be a point-to-point protocol over an unreliable medium with deterministic delays. From this perspective, SDLC does not fit easily into today's internetworking environment. The solution provided by DLS (and other internetworking devices with SDLC support) eases the transition away from today's installed base of SDLC-capable devices.

DLS Solutions

DLS includes four specific functions to solve the general problems stated above. These are discussed in detail in this article:

- Termination of the data link control at the 6611
- Storage of key LAN information in a cache in the 6611, which reduces the WAN inefficiency of LAN protocols
- Use of control algorithms that prevent the WAN from becoming too congested
- Data link conversion for accommodating older SNA devices that can support only SDLC

Local Termination

Termination of the data link control, also called local termination, link termination, or local acknowledgement, addresses three problems:

- It reduces network overhead by removing the session keepalives from the WAN
- It reduces timer expiration problems (since frames are acknowledged locally)
- It reduces the impact of the limited number of hops that a source route bridged (SRB) frame can take

SNA Needs Connection-Oriented Link

SNA requires a reliable, connection-oriented data link protocol. Once SNA presents a packet of data to the data link layer, it is the responsibility of the data link layer to ensure end-to-end delivery. For WANs, the data link control under SNA is typically SDLC while for LANs it is logical link control type 2 (LLC2). (IEEE 802 LANs can also use a connectionless data link called LLC1 if the protocols they are supporting do not need a connection-oriented data link, which SNA requires.)

Two attributes common to both these connection-oriented data link protocols are:

- Session keepalives ensure that the DLC stations at each end of the link are alive and operating
- Time constraints are placed on the acknowledgement of the receipt of a packet

Session Keepalives

Both SDLC and LLC2 make use of session keepalives in order to maintain the integrity of the connection. For SDLC, these keepalives are two-byte receiver ready (RR) frames that travel end to end between adjacent link stations. Even when there is no data to send, each pair of link stations regularly exchanges these frames to ensure the link is up.

However, when DLS transfers SNA traffic across a multiprotocol network, it encapsulates each data link frame inside a TCP/IP packet, which guarantees end-to-end delivery of the SNA data (see *SNA Perspective*, October 1991). But to maintain this logical end-to-end session across the multiprotocol network, each RR frame would be encapsulated inside a 42-byte TCP/IP packet. These 42-byte packets would be continuously traversing the multiprotocol network even when the end stations are idle, placing significant overhead on the network and impacting overall network performance.

If the user has bandwidth to burn, this might not be a problem. But, if the intermediate network consists of low- or medium-speed wide area links that most users have today, then the local termination of sessions provided by DLS will offload that unnecessary overhead and make that bandwidth available for other data traffic.

Timer Constraints

Likewise, if an end-to-end session is being maintained across a multiprotocol network, the timers at the data link control level must be adjusted to compensate for any delays encountered in the intermediate IP network. While delays in an IP network are bounded, they are sometimes much greater in duration than the end stations will accept and result in a session outage.

DLS Local Termination Solution

The DLS solution to both these problems—session keepalives impacting network performance and timer expiration affecting session integrity—lies in termination of the data link control session at the 6611. This solution in the case of an LLC2-LLC2 session is shown in Figure 4. This figure shows three concatenated sessions:

- An LLC2 session from the end device to the 6611
- A TCP/IP session across the multiprotocol network
- An LLC2 session from the 6611 to the end device

IBM terms the end-to-end logical connection through these three concatenated physical sessions a *circuit*. Even though there are three concatenated

LLC2 Local Termination

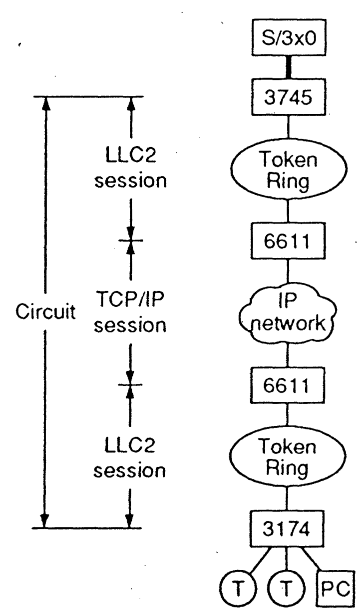


Figure 4

DLC sessions, the upper layer SNA sessions are unaware of this and are thus unaffected by it.

The 6611s with DLS locally terminate all LLC2 and SDLC sessions at the router. This means that all session keepalives (i.e., RRs) remain only on the local segment (that is, between the end station and the 6611), and the scope of all timers is only the local segment. This not only removes all the overhead traffic from the wide area network but also benefits the end systems by removing sensitivity to WAN delays.

Removing sensitivity to WAN delays is very important. With local termination of data link control connections at the 6611, no end systems making use of either the LLC2 or SDLC protocols need to have any changes in their configuration. For example, the timers in the NCP gens do not have to be adjusted to accommodate WAN delays when DLS is used. But, more importantly, this also means that the T1 timers on the thousands of token ring client stations on a large corporate network do not have to be adjusted to compensate for WAN delays. This is valuable because it is increasingly common in corporate networks for some servers to be separated from their clients by a WAN.

The only other vendor of multiprotocol routers offering this solution at the present time is Cisco with its local termination feature which it calls LLC2 Local Acknowledgment. Other multiprotocol router vendors have discussed adding local termination of LLC2 sessions to their products in the future.

Hop Count Extension

An aspect of data link termination that is offered in DLS and is unique to the 6611 is the capability to extend the hop count limitation of source route bridged (SRB) packets. The SRB specification limits the number of bridges that can be specified in the routing information field (RIF) to seven. This means that, under normal circumstances, all destination stations with which an LLC end station wants to form connections must be accessible through no more than seven bridges. If a destination is more than seven hops away, the connection cannot be established.

DLS addresses this problem by terminating the RIF at the 6611 connected to the IP network. This feature permits token ring networks on each side of a 6611 to consist of seven hops each, thus extending the reach of LLC sessions. DLS terminates the RIF at the 6611 before the SRB packets are sent across the IP network (see Figure 5).

Other vendors offer a similar capability to "extend" the hop count but each is done differently. Wellfleet routers extend the hop count by removing the RIF on every incoming packet and regenerating a new RIF on the outbound segment. Wellfleet calls this feature the Source Route Extended protocol. Cisco addresses the problem of hop count limitations by aggregating a number of intermediate routers into what it calls a virtual ring. In other words, it "solves" the problem by routing rather than bridging.

Controlling LAN Broadcasts

Most LAN protocols evolved with the assumption that the underlying medium provided nearly infinite

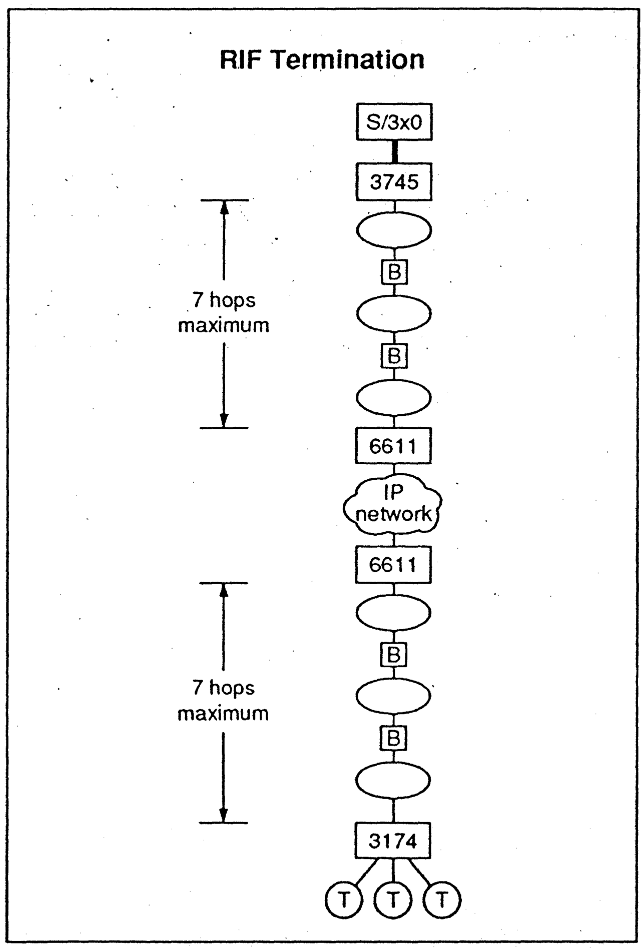


Figure 5

bandwidth. As a result, LAN protocols are very "chatty" in nature and this chattiness is often manifested in the LAN protocol with broadcast messages. Even though broadcasts, by definition, are efficient on a broadcast medium such as Ethernet or token ring, this LAN efficiency doesn't translate well into WAN efficiency since WANs are built on a series of interconnected point-to-point links.

Point-to-Point in Form; Broadcast in Intent

DLS in the 6611 has been designed to be sensitive to certain LAN broadcast messages and to perform WAN operations that are point-to-point in form but broadcast in intent. What this means is that certain LAN broadcast packets cause DLS to react by sending point-to-point directed packets to participating 6611 routers across the IP network that achieve the intended result of the original LAN broadcast packet.

A good example of this is found in the SNA session establishment sequence for token ring LANs. An SNA end station (typically a 3174 or a 3270 emulation package) sends an "all routes broadcast" packet in order to find its partner end station (typically a 3745). This packet, which is sometimes called an explorer packet or a discovery packet, must be broadcast onto all interfaces in an attempt to locate the destination station and thereby determine a route from source to destination.

When a 6611 router sees an explorer packet, it forwards it on to all other LAN segments to which it is connected (therefore acting as a token ring bridge). But it operates differently for DLS interfaces into a multiprotocol network.

CANUREACH and ICANREACH

DLS sends a special control packet called a CANUREACH to all participating remote DLS routers. Those CANUREACH packets are translated into explorer packets on LAN segments at the remote router. Once the destination is found and a response to the explorer packet is sent from the destination end station, the participating DLS routers forward an ICANREACH control packet back to the originating 6611.

The dialogue between participating 6611s in the token ring router discovery process is shown in Figure 6. The network shown in the figure is a very simple one with only one pair of cooperating DLS routers. However, it serves to illustrate the principle behind the CANUREACH/ICANREACH dialogue and its relationship to the RIF of the source route bridge protocol. Since the DLS routers must remain transparent to the source-route-capable end stations, the result of this CANUREACH/ICANREACH protocol must be the preferred route from source to destination.

6611 Route Table

While the end stations keep track of the route from source to destination by remembering the RIF, DLS-capable 6611s maintain a database of their own that aids in source routing. This database takes the form of a route table with the media access control (MAC) address as the key. This table permits a 6611 to respond directly to explorer packets seeking out MAC addresses that it knows about.

In this route table, each 6611 also maintains a *preferred* route and zero or more *capable* routes for SRB packets. In the network shown in Figure 6, there is only a preferred route and no capable (i.e., backup) routes. However, the network shown in Figure 7 on page 14 has two capable routes in addition to the preferred route. While the path through router B might be the preferred route to get to the

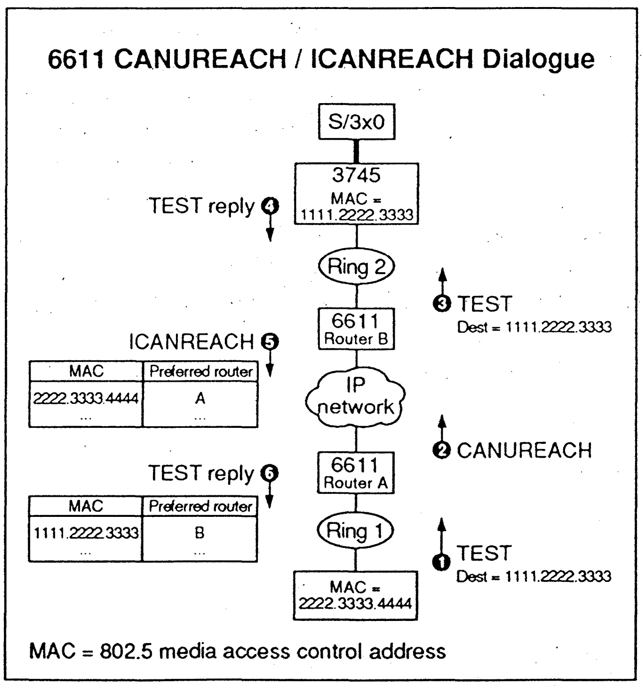


Figure 6

3745, paths through routers C and D are also capable of handling token ring traffic destined for the host. These capable routes serve as backup routes in case the preferred route is lost.

Figure 7 shows another interesting feature of DLS—reverse route information is saved in the route table of DLS routers. The information in the reply to the explorer packets is captured at the participating DLS routers and kept in the route table maintained for SRB packets. This enables any explorer packets seeking MAC addresses in the reverse direction (for example, workstation WS2 exploring for workstation WS1) to be replied to locally at the DLS router (in this example, by 6611 router B).

NetBIOS Name Caching

Another DLS feature that reduces LAN broadcasts is NetBIOS name caching. When a client workstation starts up a NetBIOS application, NetBIOS broadcasts a FIND NAME to locate the server machine. The procedure for handling NetBIOS name broadcasts is very similar to the procedure described above for token ring explorer packets.

DLS Forwards Only First CANUREACH

A final feature that reduces LAN broadcast traffic on wide area networks is the ability to handle situations following a loss of connectivity to LAN stations that are providing server-based applications.

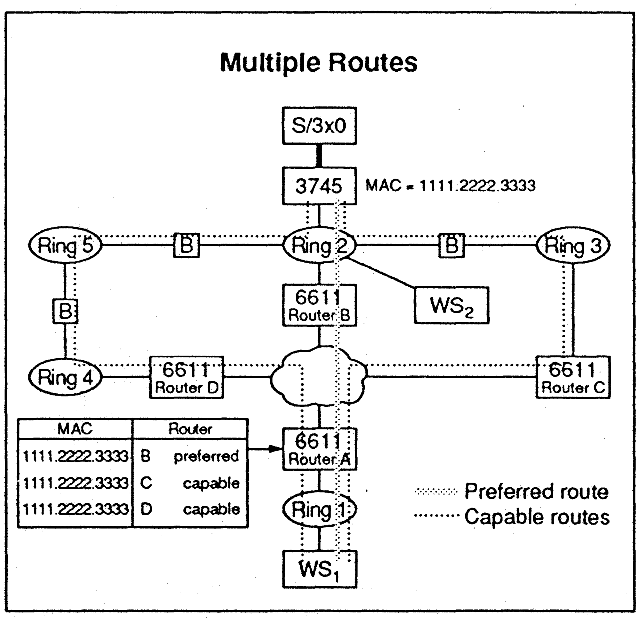


Figure 7

For example, this could happen whenever the 3745 end of a source-routed SNA network is lost; that is, when the 3745 token ring interface coupler becomes unavailable. In this case, all the SNA client workstations that were connected to that 3745, such as 3174s and 3270 emulators, would immediately attempt to reconnect by sending out explorer packets.

Rather than allowing the multiprotocol network to be flooded with CANUREACH packets, the DLS router permits only the first CANUREACH for a particular destination MAC address to be sent out. Second and subsequent explorer requests for the same MAC address are queued at the 6611 waiting for the ICANREACH reply. Once the ICANREACH reply is returned to the originating DLS router, the route table is established and all the queued explorer requests are sent a reply message.

Even though the above example illustrates the operation for explorer packets in a typical SNA network, NetBIOS has a similar problem. If a commonly used NetBIOS server goes down, a similar condition will exist on the network. DLS effectively implements a firewall for explorer storms as well as NetBIOS FIND NAME storms following link outages.

Controlling Congestion

Locally terminated sessions, such as those found with DLS, increase the need for a router's ability to control the flow of data from the end stations. The need for congestion control is more acute with locally terminated sessions because there is no direct end-to-end exchange of frame counts that make up the sending window size at the data link level. The frame counts (known as N_r (received) and N_s (sent) in both SDLC and LLC2 terminology) have local context only and therefore do not reflect what has actually been received at the remote end.

The disparity between what the local station has sent and what the remote station has received affects the number of buffers that the intermediate multiprotocol network must reserve for the circuit between those stations. This disparity is even greater when the data link types of the SNA end systems at each end of the DLS pipe are different.

For instance, an SNA end system on a 16 megabit per second (Mbps) token ring could easily overrun a 9600 bit per second (bps) SDLC link. Since the routers do not have an unlimited supply of buffers and since SNA requires guaranteed delivery of data, DLS implements a very thorough set of congestion control capabilities that throttle the end stations.

When end stations are producing data frames faster than they can be absorbed by the intermediate network, DLS exerts backpressure by issuing Receiver Not Ready (RNR) control frames at the data link control level. These RNRs have a dual effect:

- They prevent the end station from sending any more data
- They keep the data link control session alive

DLS Flow Control More Sophisticated

The flow control techniques used by DLS are more sophisticated than those found in other routers. There are two different types of backpressure mechanisms implemented in DLS:

- Flow control specific to an individual TCP/IP session
- Flow control of all participating DLS routers

The first is the flow control mechanism implemented by most of today's multiprotocol routers. The benefits of session-specific flow control are that it is simple to implement and that disjoint sessions in the routed network do not adversely affect each other during flow control situations.

The second is a more global kind of congestion control found only in DLS. A DLS router that is experiencing congestion can apply backpressure to all other DLS routers that have circuits going through the congested router. The main benefit of this kind of flow control is that the problem of insufficient buffers to handle peak loads is rectified more quickly.

On the other hand, this form of flow control also implies that data link sessions that happen to share the same DLS router pair contend not only for CPU and I/O resources on the routers but also affect each other in terms of buffer availability. Presumably, the imbalanced case of 16 Mbps token ring to 9600 bps SDLC noted above could adversely affect other circuits where the data speed on each end of the circuit is more balanced.

This more global form of DLS congestion control is shown in Figure 8. If downstream DLS router C detects a congestion problem, it sends a control message to each upstream participating DLS router (routers A and B) so they can exert the appropriate backpressure to their sending end stations.

Routers A and B in this scenario need to control the flow of only those circuits that are transported over the network to the congested router C. Additionally, since RNR applies only to the data link control sessions, only those link stations destined through the congested router are affected. All other link stations, possibly including other link stations within the same workstation, are unaffected by the throttling mechanism.

As with all congestion control mechanisms, DLS has the ability to get out of the throttled state. Once the congestion is alleviated, the appropriate DLS routers are notified and data is allowed to flow normally again.

Accommodating SDLC Devices

Discussion has so far focused primarily on LLC2 sessions on token ring networks. However, many

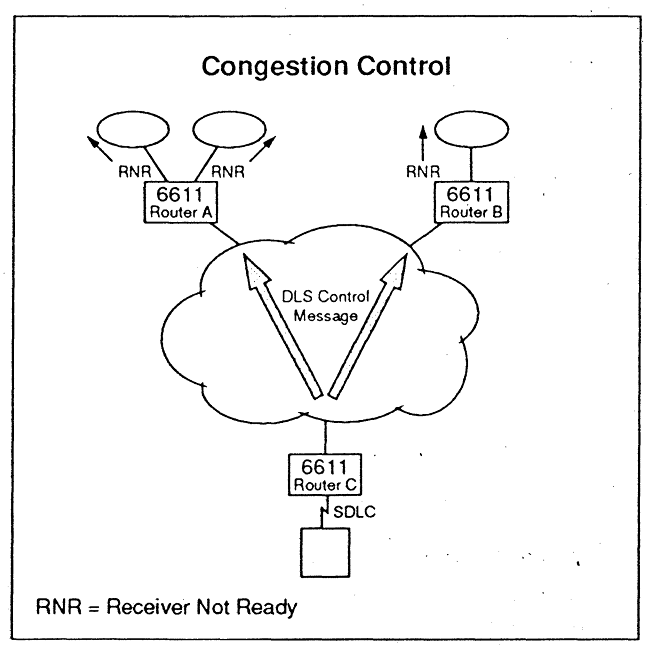


Figure 8

devices, such as the older 3274 cluster controllers, are not capable of connecting to token ring networks. IBM has addressed this problem by accommodating SDLC-attached physical unit 2 (PU 2) devices within the DLS architecture.

SDLC-to-LLC2 Conversion

Since both data link control types share a common ancestry related to ISO 3309's High-level Data Link Control (HDLC) protocol, the translation between LLC2 and SDLC is a fairly direct process. Mapping control frames and information frames from one data link type into those required by the other is very straightforward. Conversion from SDLC to LLC2 can be found today either in standalone products—the SDLC Link Server from Netlink of Raleigh, North Carolina, and the SNA Network Access Controller for Token Ring (SNAC/TR) from Sync Research of Irvine, California—or as an option in multiprotocol routers from both IBM and Cisco.

DLS implementation of SDLC-to-LLC2 conversion permits downstream SDLC-attached PU 2 devices, such as 3274s, 3174s, and AS/400s, to be connected into serial ports of the 6611 router and emerge on the host side appearing as if they were attached on a token ring. As shown in Figure 9, while both LLC2 and SDLC are supported on the remote side, DLS only supports LLC2 on the host side. That is, a 6611 can act as a primary but not a secondary SDLC link station. Router B in Figure 9 would act as the primary SDLC link station with the 3274, terminate the SDLC link, and send the SNA traffic to Router A, which would have no awareness of or involvement in the SDLC session. As explained in last month's Architect's Corner of *SNA Perspective*, supporting only LLC2 on the host side is not necessarily a significant restriction, primarily because users willing to put their SNA traffic on LANs would also likely have attached their 3745 to the LAN.

One of the few drawbacks to this conversion is that it affects the sysgen. Since the downstream SDLC devices will appear as token ring–attached devices, the sysgen needs to change to accept their appearance as token ring (i.e., switched major nodes) rather than link-attached SDLC devices (i.e., switched or nonswitched SDLC nodes).

Only one of the SDLC-to-LLC2 conversion products on the market today offers SDLC support for node type 2.1 devices. Sync Research recently added this capability and Netlink has made a statement of direction to support node type 2.1 links in the future. There are two good reasons why downstream node type 2.1 support is rare:

- A significant amount of extra configuration information is required to support the extended Exchange Identifier (XID) for type 2.1 nodes
- Almost all type 2.1 nodes that are SDLC today are also capable of supporting token ring connections

Even though today's DLS implementation in the 6611 router supports only LLC2 and not SDLC on the host side, the DLS architecture is flexible enough to handle SDLC on the host side as well. IBM probably chose not to implement it because of the lack of perceived market demand.

Qualified Logical Link Control (QLLC)-LLC2 conversion could be a viable addition to DLS. This would allow SNA devices with an X.25 interface that would normally connect to a 3745 across X.25 through NPSI and QLLC to connect instead with the

DLS Data Link Possibilities for SNA

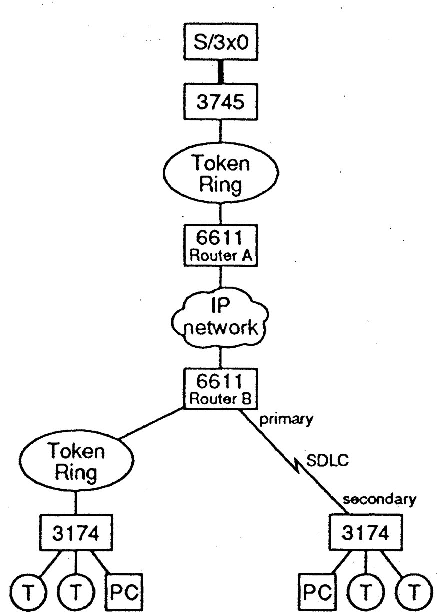


Figure 9

host through a LAN environment. Sync Research is the only vendor whose product offers this capability, and its product is also resold by McDATA as the LinkMaster 7200. The DLS architecture could also support this and, though IBM has obviously chosen not to offer it at this time, this is among the future enhancements *SNA Perspective* expects for the 6611 in the long run.

DLS Disadvantages

Network Management

The 6611 cannot be managed by IBM's LAN Network Manager, nor do we expect it to be (see *SNA Perspective*, June 1992). It can be managed by SNMP monitors such as NetView/6000 on the RS/6000 and thus, indirectly, by NetView. Since the 6611 is more of a multiprotocol router than a source route bridge, however, the SNMP management is more important.

No Standard

One problem with DLS is common to most internetworking devices like the 6611—incompatibility. These products have emerged, in part, to address the incompatibility between different protocols but, in such a new and rapidly growing market, these "solutions" themselves were often incompatible. DLS is yet another new approach which, as discussed here, offers several advantages and yet is also incompatible with other offerings on the market. It should be noted that this incompatibility issue is an edge router phenomenon; intermediate IP routers are interoperable.

No SNA "Routing"

The 6611 with DLS does not have the capability to route IBM's own subarea SNA that the 3745 provides in software implementing SNA PU 4. However, the complexity of full SNA "routing" would require much more expensive hardware and software than would be competitive in the bridge/router market. (This is, in part, because SNA routing involves higher levels than other protocols and cannot be routed strictly at layer 3—which is why it is often called an unroutable protocol.) This topic was discussed at length in *SNA Perspective*, July and August 1991.

In early 1991, Cisco announced it would provide PU 4 routing on its routers, but now says it will provide only two features of PU 4: transmission groups and class of service. IBM has not said whether the 6611 will be enhanced to include these or other selected PU 4 capabilities.

DLS and the Competition

IBM has not made the DLS technology in the 6611 available as an openly licensed product. If it were licensed, vendors of similar networking products could readily have implementations of DLS that are compatible with the 6611. At the present time, the only networking vendor that, as an IBM business partner, has been given complete DLS specifications is Network Equipment Technologies (N.E.T.) of Redwood City, California. *SNA Perspective* believes that IBM is not currently seriously considering commercially licensing DLS or proposing it as a standard.

However, IBM has said it is in the process of documenting the operation of DLS for the purpose of offering it as an information-only Request For Comments (RFC) to the Internet Engineering Task Force in the near future. The information-only RFC is not comparable to publishing DLS but rather provides only the information necessary to interface with it. This step will further enhance the interoperability of other vendors' implementations with those on the 6611.

There are a number of products on the market that provide services similar to those found in DLS. Table 2 on page 20 identifies a number of SNA and NetBIOS-specific features that are found in the multiprotocol router and data link conversion marketplace today. Most, but not all, of the features are found in DLS.

DLS versus APPN

By being sensitive only to the data link control layer of SNA, DLS can accommodate all forms of SNA traffic from subarea flows (i.e., PU 4-to-PU 4) to

(continued on page 19)

Architect's Corner

Protocol Engineering —Three Futures

by Dr. John R. Pickens

Now that IBM has tipped its hand for the near term with licensable APPN, APPN/MVS, APPN on the bridge/router, APPN+, etc., let's step back a bit and ask an open question: "What's next? How will protocols evolve, both inside and outside of the IBM environment?"

The answer is: "This is a good question."

During the INET'92 conference in Japan, I had an opportunity to discuss the general subject of protocol futures with several colleagues who are devoting a major chunk of their time and energy to the issues of protocol engineering.

Good Enough?

The first question many ask is, "Aren't the current protocols good enough?"

Taking this position is very appealing, especially to vendors who are seemingly hijacked into implementing every protocol. And especially to users, who are bending under the burden of training and drowning in the conundrum of complexity. Many people would say, "Current protocols are good enough. Let us understand and master what we have today."

There is only one problem with this view. The technical environment is changing, and rapidly—100-MBps FDDI, 150-MBps to 600-MBps (and above) ATM/STM, and gigabit networking infrastructures are on the horizon. New requirements are emerging for multimedia synchronization and for tighter coupling between applications and the underlying communications environment. Believe it or not, protocols designed for the bandwidth and delay characteristics of today's infrastructures do not scale well in these emerging environments.

I see three camps, at least, for protocol evolution—the universalists, the fundamentalists, and the revolutionaries. (Note: only a brief summary is given here of each camp—each certainly deserves more extensive treatment at some future date.)

The Universalists—Simplify, Simplify

The universalists are concerned less with fundamental protocol evolution and more with simplification of the infrastructure—from the perspective of users and applications. Two good examples of this trend are:

- The multiprotocol transport network (MPTN) initiative submitted by IBM to X/Open [*Introduction to Multi-Protocol Transport Networking (Revised)*, April 17, 1992, IBM Research Triangle Park]
- The virtual Internet Protocol being prototyped by Sony corporation [INET'92 Proceedings].

The IBM work addresses the problem of protocol and protocol interface translation (sometimes called middleware). The MPTN initiative is very ambitious. With only small loss of function, MPTN makes it possible for users to install a single interface (e.g., CPI-C, sockets), run that interface atop any protocol substructure (e.g., TCP/IP, SNA), and interoperate with any other protocol structure via gateways (e.g., TCP-to-SNA translation). Ambitious? Yes. Realizable? Probably. Loss of function? Small but probably livable.

The Sony work is less ambitious and takes a more canonical approach. That is, it defines a single virtual network service layer and hides the differences between underlying realizations beneath that layer.

Can the universalists succeed? I think the jury is still out on this one but limited success is almost assured. The benefit delivered? Simplification. The weakness? Efficiency. It is tough enough to maximize performance with a single consistent protocol family. It is harder to marry dissimilar protocol families in this way.

The Fundamentalists—Keepers of the Flame

The fundamentalists are keepers of the status quo, the heroes of the user community, in a sense. Their mission is to maintain current protocols and architectures with minimal (i.e., incremental) change. A good example of this trend is the work underway to extend TCP for high-speed networking—selective acknowledgments, larger windows, open-loop round-trip-time-estimating algorithms [IETF work in process].

APPN+ will probably be another example of this approach.

The strengths? Maintains a semblance of stability—"minimal" change to the protocol and protocol API environment. The weaknesses? The revisions are really too substantial, so can the protocol still be called by the same name? Further, the changes, well-intentioned though they are, may not go far enough to meet the needs of the next generation of high-speed networking.

The Revolutionaries—Do it Right This Time

The revolutionaries are dealing with a blank slate. They revisit the requirements of networking as reflected in the fundamental new technologies and create new models and mechanisms. An early example of this approach is the NetBLT protocol [IETF RFC998, 1987] which utilizes new paradigms for high-speed networking like rate-based flow control and selective acknowledgments.

The strengths? Development of protocols optimized for the new infrastructure of high-speed, high-latency networking. Also, extending the service model to include multimedia synchronization primitives, etc. The weaknesses? Lack of standards, deployment to operating systems, communications systems.

Inevitable Conclusions?

It should be manifest by now that the dual objectives of (1) stabilizing (and standardizing) the protocol environment and (2) evolving protocols toward fulfilling the needs of new high-speed internet infrastructures are, yes, contradictory goals.

No one, neither vendor nor user, should expect stabilization in the protocol world, at least in the forseeable future.

So, I ask, "Can we afford to have yet another protocol?" And answer, "Can we afford not to have another protocol?" To which, in closing, I would add that all three camps—the universalists, the fundamentalists, and the revolutionaries—are talking about new protocols. ■

(continued from page 17)

peripheral node flows (i.e., PU 4-to-PU 2). (Note: PU 4-to-PU 4 is currently supported only for LLC2-to-LLC2 communication.) Since DLS is merely extending the range and scope of data link services, questions of scalability must be raised. And if DLS is not a scalable alternative, would an APPN Network Node router provide a better solution?

DLS should really be viewed as a complementary alternative to APPN routing. While DLS might not be able to maintain thousands or even hundreds of circuits efficiently, APPN was designed to do just that. The amount of conversion, twiddling, caching, and processing of data link control information that DLS performs places a real upper limit on the number of data link connections traversing through the 6611.

(continued on page 20)

(continued from page 19)

But as mentioned above, DLS is insensitive to the type of SNA traffic it is handling. Until APPN readily accommodates other forms of SNA traffic besides LU 6.2, the same statement cannot be made of APPN routing. Fortunately, the location in a customer's network of 6611s with DLS today will most likely be the exact location where the customer would like to place an APPN network node in the future.

A final benefit that APPN has over DLS is the granularity of SNA entities that can be routed. Since DLS switches data links, only PUs are routed. With APPN, individual LUs can be routed.

Summary

DLS is a solution today that preserves a customer's investment in its existing protocols. DLS does this by efficiently accommodating SNA and NetBIOS traffic in a multiprotocol network. It reduces the WAN inefficiency of LAN protocols and supports nonLANable SDLC devices. Some of the more significant DLS tools are local termination, congestion control, and broadcast control. DLS adds no new capabilities to the internetworking market, but implements several features in a new way that *SNA Perspective* believes is worthy of the user's consideration.

Many capabilities of DLS are arguably "bandaids" for protocols that are being used for purposes outside their original design. Of course, the same can also be said for many features of other multiprotocol bridge/routers. The current marketing phrase for this support is "protecting customer investments."

Since DLS merely extends the range and scope of data link services, it was not designed to support very large networks. *SNA Perspective* considers DLS is a credible option for accommodating SNA in multiprotocol networks until full APPN routing becomes available. ■

Router SNA and NetBIOS Feature Comparison

Property		IBM	Cisco	Proteon	Wellfleet	Sync Rsrch	Netlink
Data links supported	LLC2-to-LLC2						
	Passthrough	No	Yes	Yes	Yes	—	—
	Local termination	Yes	Yes	No	SOD	—	—
	RIF termination	Yes	Virtual ring	No	SR extended	—	—
	SDLC-to-SDLC						
	Passthrough	No	Yes	Yes	Yes	—	—
	Local termination	No	9/92	No	No	—	—
	SDLC-to-LLC2 (sec-pri)						
	Downstream PU 2.0	Yes	Yes	No	Yes (3rd party)	Yes	Yes
	Downstream NT 2.1	No	No	No	No	Yes	SOD
	Same router conversion	Yes	Yes	No	No	—	—
	LLC2-to-SDLC (sec-pri)	No	9/92	No	No	Yes	No
	QLLC-to-LLC2	No	No	No	No	Yes	No
Broadcast control	NetBIOS name caching	Yes	9/92	4Q92	SOD	—	—
	Proxy explorer	Yes	Yes	No	SOD	—	—
	Explorer firewalls	Yes	No	No	No	—	—
Congestion control	Per TCP/IP session	Yes	No	No	SOD	—	—
	All participating routers	Yes	No	No	No	—	—
Network management	LAN Network Manager agent	No	Yes	No	SOD	—	—
	Native NetView support	No	No	No	No	Yes	Yes
	SNMP agent	Yes	Yes	Yes	Yes	9/92	No

Table 2