SNA Perspective

Volume 13, Number 11
November, 1992
ISSN 0270-7284

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

Old Apps, New Nets—Dependent LUs Across APPN Part I: Existing Support

Although most data centers acknowledge the need for an open network strategy, approximately 80 percent of users continue to rely on dependent LUs through 3270 terminals, 5250 terminals, banking and point-of-sale terminals, and PCs with terminal emulation through controllers, protocol converters, and PC boards for micro-mainframe connectivity. Applications migrate more slowly than networks.

Current subarea SNA customers who are considering options for moving to APPN want to know how their existing workstations and applications will be supported. Dependent LU traffic depends on hierarchical SNA network support. However, products based on APPN, IBM's flagship peer networking architecture, today support APPC almost exclusively and not dependent LUs.

This article discusses currently available and announced IBM dependent LU support over APPN including VTAM and the AS/400. To provide context, we note the size of the existing installed base, discuss ten decision criteria for selecting a solution for dependent LUs over APPN, review several LU-related concepts, and analyze the three main technical reasons why supporting dependent LUs on APPN is such a challenge.

(continued on page 2)

Part II: Future Support

In order to plan for both current and future investments, users need to know how their dependent LUs will be supported over APPN. *SNA Perspective* has interviewed IBM and several other vendors to understand their directions. Here we present to our readers the most detailed description, comparison, and analysis available of the emerging products in this arena.

In this second part of this two-part series, we address two approaches—IBM's dependent LU server/requester model and encapsulation of dependent LU traffic in APPC. We discuss each approach, including three techniques for encapsulation, and analyze its expected strengths and weaknesses. We compare them by ten decision criteria the user will need to take into account. We conclude with recommendations to users regarding continued investments in dependent LUs and planning for the future.

(continued on page 9)

In This Issue:

Considering the move from subarea SNA to APPN? Concerned about keeping your workstations and applications? Confused about why APPN has problems with dependent LUs? Wondering whether or how these problems will be addressed by IBM and other vendors? *SNA Perspective* devotes this entire issue to these important questions.

Our architect discusses the value of using a compass, rather than an array of maps, to navigate the currents of the SNA environment.

(continued from page 1)

It is not the intention of this article to judge or justify the value or logic of subarea SNA and dependent LUs or to debate the wisdom or value of migration to APPN. Instead, *SNA Perspective* takes a pragmatic approach to the situation and presents information to enable SNA users to understand the issues involved in supporting dependent LUs over APPN.

This two-part series concerns *routing* of subarea SNA traffic. However, it should be noted that an APPN node with bridging support, such as the 6611, can *bridge* SNA traffic across APPN. For some users, a bridged solution may be adequate for part or all of their network.

The Size of the Situation

By 1990, the number of PCs installed worldwide outnumbered terminals. Further, PC shipments each year far outstrip the number of terminal shipments, and many terminals are being replaced by PCs.

However, millions of terminals and PCs are being installed and use dependent LU communications today.

About ten million workstations worldwide currently access 3270 applications. (This number has been extrapolated from market research data by Dataquest of San Jose, California, and International Data Corp. of Framingham, Massachussetts.)

Of those ten million workstations, some seven million are connected to about 850,000 3270 controllers, in addition to about two million printers. About half of these workstations are 3270-type terminals from IBM or compatible products from companies such as Memorex/Telex of Irving, Texas. The other half are PCs using 3270 interface cards from companies such as DCA of Alpharetta, Georgia.

Approximately two million PCs access 3270 applications through 3270 LAN gateways from vendors such as Attachmate of Bellevue, Washington. Almost another million workstations access 3270 services through protocol converters.

Not Just 3270

These ten million 3270 workstations do not include the many controllers, personal computers, terminals, printers, and emulators using 5250 protocols.

The number also does not include the many banking and point-of-sale systems that are based on dependent LUs.

Hosts

The applications used by these workstations are found on about 60,000 mainframes around the world and several hundred thousand S/3x and AS/400 systems.

Users Will Not Discard the Old to Embrace the New

Many subarea SNA users are interested in upgrading their networks to support Advanced Peer-to-Peer Networking (APPN) because of its distributed, dynamic, and adaptive capabilities. But customers are not inclined to easily discard their dependent LU systems in order to move to any new technology, regardless of the wonders it may offer. They want to know how APPN will support them.

It is not simply a matter of upgrading dumb terminals. Users do not want to discard or even significantly change their existing investment in applications, workstations, controllers, interfaces, cabling, and user training.

Certainly, over time, many of these investments will be migrated, and IBM is striving to provide an increasing array of migration tools to ease this transition. But, realistically, few users will attempt a hard cutover from a subarea SNA environment to an entirely APPC/APPN-based environment. Nor do they want to maintain a dual network for subarea and APPN SNA. Therefore, dependent LUs must be supported across APPN.

IBM Plans

Although the APPN architecture was designed from the start to allow dependent LUs, several products may not have implemented these features. Examples of missing support *SNA Perspective* is aware of include *start data traffic* and *clear* RUs, RU size=0, and fixed pacing. These APPN implementations must be upgraded to support dependent LU traffic.

IBM will probably provide no-charge fixes for these shortcomings in its products. Otherwise, an intermediate node could bring down the session. The initial APPN licensed network node code (scheduled to be available in the first quarter of 1993) will support these basic capabilities.

IBM has completed its internal architectural plan for mapping subarea capabilities to be implemented in APPN. However, there is currently no published information from IBM on dependent LU support over APPN, either current capabilities or future plans.

Decision Criteria for Dependent LU Support over APPN

Investment protection
Host applications, user interface, 3270 controllers and terminals (IBM and non-IBM), coax adapters and cabling, 3270 emulators (standalone and LAN-based), 3270 protocol converters.

Implementation effort/cost
Ease or difficulty of installation and configuration; money, time, and retraining.

Performance
Throughput and overhead.

Leverage
Multiple benefits reaped from any additional investment.

Host independence
Possible without VTAM (such as an AS/400 network) or possible with a noncurrent release of VTAM.

Manageability
Visibility and accessibility to NetView and VTAM management.

Scalability
Support for large networks.

Interoperability
Ease of associating with TCP/IP.

Extensibility
Ease of expanding and adaptability to support newer technologies.

Portability
Ability to port code easily to a range of platforms.

Table 1

Solutions: Native or Encapsulation

There are two basic approaches to routing dependent LUs across APPN: native support and encapsulation.

IBM's strategic solution is a native approach, which will be implemented in two steps: first, support for dependent LUs that are configured as they are today; second, support for dependent LUs attached arbitrarily to any point on the APPN network. This second step is referred to as the dependent LU server/requester (dLS/R) model.

There are also several proponents, inside and outside of IBM, for one of several encapsulation techniques.

Ten Decision Criteria

SNA Perspective expects that both approaches will be implemented by IBM and other vendors, so users will have the opportunity to choose.

In Table 1, we list ten criteria to consider when comparing native support and encapsulation. These criteria are used in part II to compare the dLS/R model and the encapsulation techniques.

Basic Challenges

There are three basic challenges in supporting dependent LUs over APPN:

- SSCP sessions and APPN
- Routing information and dependent LU BINDs
- APPN secondary LU initiate

SSCP Sessions and APPN

The first basic challenge to supporting dependent LUs on APPN is that dependent LUs need SSCP-LU and SSCP-PU sessions but APPN does not support SSCP sessions. (See the sidebar "LU Review" on pages 21 and 22) Therefore, a solution must somehow support (through encapsulation) or simulate (spoof) these SSCP sessions in APPN.

It also means that, if the SSCP sessions are simulated, many actual SSCP services must still be provided in some way. For example, network management and other information that usually flows on the SSCP sessions must be captured and delivered to the appropriate place. If not, the APPN solution will degrade existing functionality.

Also, a resource on the network should provide the search and session initiation support capabilities the dependent LU expects from SSCP. Otherwise, its destinations must be preconfigured.

Routing Information and Dependent LU BINDs

The second basic challenge is related to routing.

In subarea SNA, a peripheral node depends on subarea nodes (PU 5 or PU 4) to find the desired partner, select the route, and route traffic. Therefore, its BIND does not include routing information.

In APPN routing, on the other hand, each network node is responsible for routing. This is done by way of route selection control vectors (RSCVs). The network node finds a target application by sending a locate request. The returned responses include the name of the network node serving the located destination and the tail vectors from that network node to the end node containing the application. From this information and its topology database, the source network node then calculates the optimal route, which it implements in an RSCV.

Independent LU BINDs are designed to include this RSCV, while dependent LU BINDs are not. Therefore, to run on APPN, either the dependent LU BINDs must be adapted to include an RSCV or the dependent LU session must be encapsulated in an independent LU session. The former is the technique used by the dLS/R model, while the latter is an encapsulation approach. *SNA Perspective* does not know whether the licensed APPN network node code includes the capability to extend a dependent LU BIND to include the RSCV.

Another related consideration is session priority. An additional vector that independent LU BINDs contain is a class of service/transmission priority field (COS/TPF). For a dependent LU BIND, the transmission priority was added by subarea nodes in creating an extended BIND. The COS for each subarea LU session pair was preconfigured in the host and used to select the virtual route. Over APPN, the network node must be given to the COS to use in its RSCV calculation or it could default to a certain COS for all dependent LU traffic. As with RSCVs, dependent LU BINDs will probably be extended to include a COS/TPF. It is unclear at this time how this will be done and whether or when this support will be included in the licensed APPN network node code.

APPN Secondary LU Initiate

A third basic challenge is that APPN nodes and their applications do not currently support APPN secondary LU initiate. Secondary LU initiate is the ability of an application node to respond with a BIND to a session initiation request from a secondary LU. This is not needed, of course, in environments where all dependent LU sessions are initiated by the application—primary LU initiate.

A dependent LU always acts as the secondary LU in a session, which means that it does not generate the BIND to establish the LU-LU session. The dependent LU can only request its SSCP to direct the primary LU (usually a host application) to generate a BIND. (This is analogous to the era when a lady could not ask a gentleman to dance, but could indicate to a third party such as her chaperone that she would be willing to dance if asked.)

Secondary LU initiate has been supported since early in the history of subarea SNA. It is sent in initiate (INIT) if the application is on the same host or in cross-domain initiate (CDINIT) if the target application is not on the same host as the SSCP owning the dependent LU. INIT and CDINIT both flow on SSCP control sessions.

But, as noted earlier, APPN does not support SSCP sessions. Further, independent LUs over APPN do not need secondary LU initiate. This is because an independent LU sends its session establishment request in the form of a BIND. Therefore, there has been no APPN command to request an application node to generate a BIND.

APPN secondary LU initiate provides this capability. It is a function of the APPN node, not the application itself. The directed search will contain the dependent LU BIND image, the LU's location information, and the secondary LU initiate parameter.

Both the workstation node and the application node must support secondary LU initiate. Other intermediate nodes on the path need not support it. In a concatenated network, any interchange or border nodes must also support secondary LU initiate.

Product Support: Current and Future

The remainder of this article examines current and announced support for dependent LUs over APPN. Part II discusses future support in light of expected strengths and weaknesses.

- Current
  - VTAM 3.2 and the composite LEN node
  - AS/400—5494 and display station passthrough
- Announced
  - VTAM 4.1 and the composite network node
- Future
  - dLS/R
  - Dependent LU encapsulation possibilities

VTAM 3.2 and LEN Node

In 1987, ACF/VTAM Version 3 Release 2 and ACF/NCP Version 5 Release 2 were announced. VTAM 3.2 supports LEN node and can work with NCP 5.2 as a composite LEN node.

Today, this LEN node cannot support control sessions for dependent LUs across APPN. VTAM LEN node can support the application side but, until the dependent LU side is supported with VTAM 4.1, this capability is irrelevant. Once VTAM 4.1 is available, a VTAM LEN node could serve as an application node for dependent LUs. Even then, the constraints are:

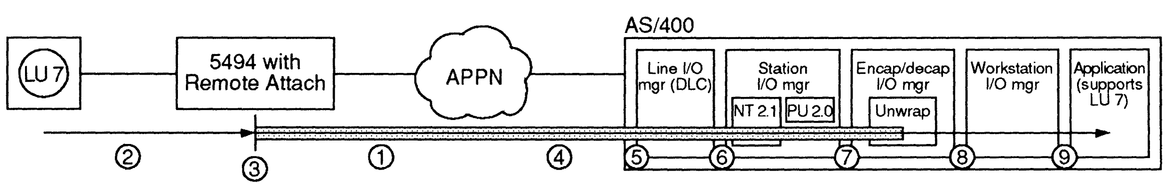
- The application must initiate the session. Secondary LU initiate is not supported.
- The relationship between the application and the dependent LU must preconfigured because the LEN node has no search capabilities.

AS/400

5494 Remote Controller

OS/400 Version 2 Release 1.1 supports the 5494 Remote Controller, a new controller in the 5250 family. A PS/2- and OS/2-based platform, the 5494 is a type 2.1 LEN node and supports APPC.

AS/400 Dependent LU Encapsulation with 5494



① AS/400 has APPC session with controller for SSCP-LU sessions, and an APPC session for each LU

② Display logs on, sends data

③ Controller program intercepts data

④ Controller encapsulates data and sends it in APPC session. (Data is in LU 6.2 frames with general data stream code identifying LU 7 user data.)

⑤ LU 6.2 frame is sent into AS/400 line I/O manager

⑥ Line I/O manager sends to station I/O manager (analagous to PU)

⑦ Station I/O manager sends to encapsulator/decapsulator I/O manager

⑧ Encapsulator/decapsulator unwraps LU 6.2 and sends data to workstation I/O manager

⑨ Workstation I/O manager sends data up stack to application

Note: SSCP-LU flows handled similarly to data.

Figure 1

Existing 5394 controllers can be upgraded to support 5494 capabilities.

Among the 5494 features is a mechanism for encapsulation of LU types 4 and 7—the LUs for 5250 printers and terminals for S/3x and AS/400—across an APPN network between the controller and its host (see Figure 1 on page 5). Since it is a LEN node, the 5494 must be logically adjacent to an APPN network node server, though it could be arbitrarily connected with regard to its owning AS/400.

One APPC session between the controller and the AS/400 carries the SSCP-PU session and all the SSCP-LU sessions for the controller. Although for simplicity they are not shown in the figure, there is also a separate APPC session for each dependent LU to carry the LU-LU session.

As shown in Figure 1, the controller intercepts and encapsulates the data in its APPC session with an AS/400. The encapsulation/decapsulation code on the controller and the AS/400 are paired through an architected TP name.

This is not a dynamic pairing—it is preconfigured in the controller and the AS/400. The 5494 and all its LUs are configured to communicate with a particular AS/400.

However, the target application can be on another AS/400 or S/370/390. To access such an application, the dependent LU sends a *start passthrough* and then specifies the target application. The controller-owning AS/400 would find the application and, if on APPN, establish a second APPC session between itself and the host with the target application. All data traffic from the controller flows first through the owning AS/400 and from there to the application host. If the target application is a 3270 application on an S/370/390 host, the owning AS/400 also provides 3270 emulation support.

This passthrough could add overhead to the owning AS/400 and also requires two APPN paths. However, the approach has the advantage of simplicity. The encapsulation code on the controller is about 2 KB and the code on the host is a similar size. This indicates that a basic encapsulation technique for dependent LU support need not be a large piece of code.

Display Station Passthrough

The AS/400 since OS/400 Version 1 Release 1 and even the earlier S/36 and S/38 have supported display station passthrough, an earlier encapsulation approach. For display station passthrough, a 5250 controller, which has no APPN capability, must be adjacent to its AS/400 or S/3x host. Through the host, it can communicate across an APPN network. Only the data stream and header record is encapsulated; the LU 4/7 control sessions are not. Selected control information is mapped into APPC.

VTAM 4.1 and the Composite Network Node

IBM will implement support for dependent LUs over APPN in two steps. The first step is implemented in the APPN network node in VTAM 4.1 which is scheduled to ship in the first half of 1993. It supports dependent LUs that are configured as they are today for subarea SNA. The second step, discussed in part II under "Dependent LU Server/Requester Model," will support dependent LUs attached arbitrarily to any point on the APPN network (see Figure 2). This first step provides equivalent functionality over APPN as over subarea SNA, but does not provide much additional benefit to dependent LUs.

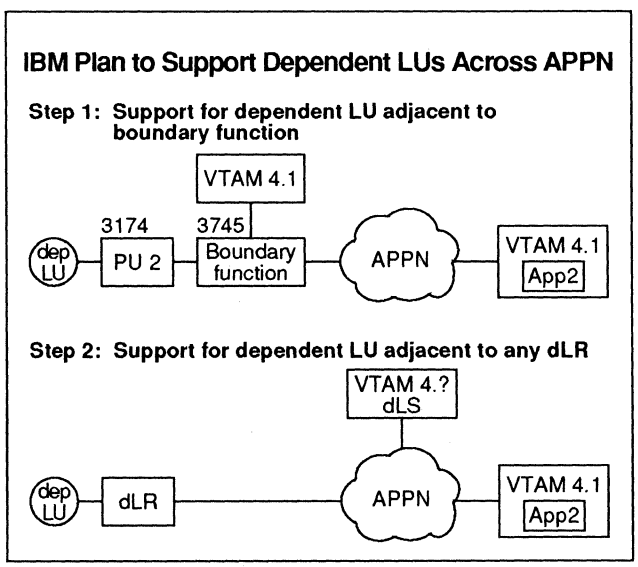


Figure 2

VTAM 4.1 can be configured with one or more 3745s with NCP 6.2 to form a composite network node. However, its dependent LU support does not require a 3745. For example, the peripheral node could be locally attached to the host or could attach across a LAN through a 3172.

With VTAM 4.1, dependent LU traffic for LU 0, 1, 2, and 3 is supported natively across APPN. An illustration of VTAM 4.1 dependent LU support is shown in Figure 3. There are three constraints:

- The peripheral node supporting the dependent LU must be logically adjacent to its boundary function.
- That boundary function, if on APPN, must be part of an APPN network node (VTAM 4.1).
- The target application node may require APPN secondary LU initiate (VTAM 4.1) .

Logically Adjacent

In subarea SNA, all currently installed peripheral nodes such as a 3174 or a 3270 LAN gateway must be logically adjacent to their boundary function, either in the host or communication controller. A logically adjacent connection can be through a hard-wired connection, an SDLC link, an X.25 link, a LAN connection through a gateway, or any other configuration that subarea SNA supports today.

This logical adjacency is the only requirement for the peripheral node. It does not need to contain any APPN support, either network node or end node. Even if a 3174 has network node installed, it is not used for dependent LU support under VTAM 4.1, as illustrated in Figure 3. Any dependent LU type 0, 1, 2, or 3 device or software at any revision level from any vendor supported today over subarea SNA by its boundary function should automatically and with no change be able to communicate over APPN through that same boundary function.

Boundary Function Part of Network Node

The boundary function supporting the dependent LU must be configured as part of a VTAM 4.1 network node. Alternately, the dependent LU's host could access APPN indirectly through subarea-APPN interchange node.

The SSCP-PU and SSCP-LU sessions are supported exactly as they are today—since the SSCP is in the host owning the boundary function, the control sessions do not enter the APPN network.

Secondary LU Initiate

The application and application subsystem do not need to be altered in any way for VTAM 4.1 APPN support. The application need only be defined in VTAM as an independent LU to be accessed over APPN.

Dependent LU Support over APPN with VTAM 4.1 and NCP 6.2

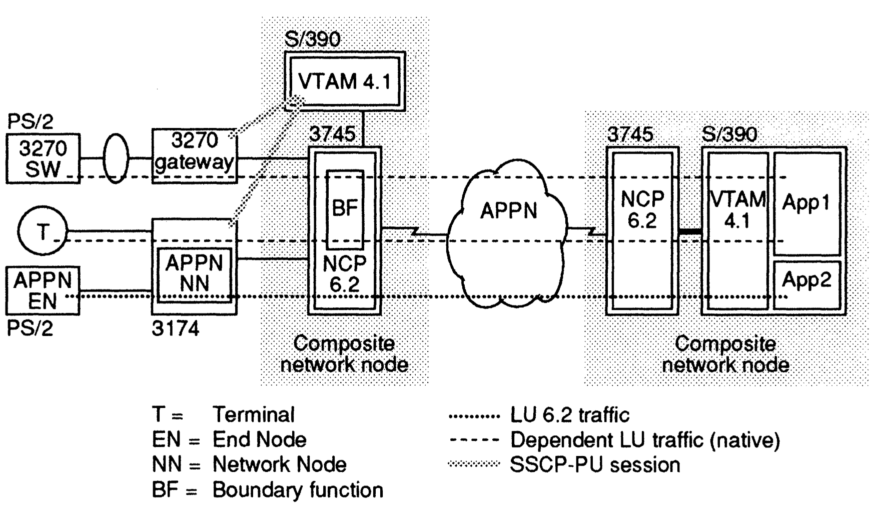


Figure 3

However, if the target application resides on an APPN network and the customer wants dependent LUs to be able to initiate sessions, the VTAM owning the application must support APPN secondary LU initiate. APPN secondary LU initiate, discussed earlier in detail, is implemented in VTAM 4.1.

IBM has not said when or whether it will support APPN secondary LU initiate in the AS/400.

The first release of the licensed APPN network node code, scheduled to ship early in 1993, will not support secondary LU initiate. This means that licensees cannot use this code to support dependent LUs either at the peripheral or application side for sessions initiated by secondary LUs. IBM says that secondary LU initiate is a logical candidate for inclusion in a future release of the licensed code.

How It Works

Using a simple example, both the dependent LU and the application are on separate hosts on the same APPN network and have VTAM 4.1.

The owning host and the dependent LU go though the usual subarea process for starting SSCP-PU and SSCP-LU sessions, none of which flow over APPN. The dependent LU requests a session with a remote application. The owning host locates the application and sends a directed search containing a BIND image and a secondary LU initiate parameter.

The network node supporting the target application on VTAM 4.1 calculates the optimal route to the boundary function node of the dependent LU. It then takes the BIND generated by the application and creates an extended dependent LU BIND containing an RSCV. The BIND is sent natively across APPN.

The boundary function will remove the RSCV and deliver a standard dependent LU BIND to the dependent LU. After the BIND, the dependent LU data ride natively on the APPN network between the boundary function and the application node.

Multiple Networks

The above is a simple example, with both the workstation and the application on the same APPN network. There may be intervening networks. To support dependent LUs, only the nodes on APPN will need VTAM 4.1.

For example, the workstation could be on a subarea SNA network and the application on an APPN network. In this case, an interchange node between the subarea and APPN networks is required. VTAM 4.1 is the first release that supports this interchange node. This workstation host could be at any VTAM release level.

Alternatively, the workstation and its host could be on an APPN network while the application is on a subarea network. Again, an interchange node is required. The application host, in this case, could be at any VTAM release level.

Today, support is not provided for a workstation and an application on two separate APPN networks (i.e., two separate NETIDs). Connecting two APPN networks requires a border node. *SNA Perspective* expects VTAM 4.2 will support border node. The AS/400 supports a basic type of border node that could possibly support primary LU initiated sessions, but it does not support secondary LU initiate.

Summary

More than ten million workstations use dependent LU sessions to access applications today. Users will not discard them to begin migration to APPN. Therefore, APPN must support dependent LUs.

There are two main approaches to dependent LU traffic across APPN—native and encapsulation. Current implementations of each approach were discussed in this article. Future directions for each approach, including the dLS/R model, are examined in part II.

Ten criteria should be used in deciding which approach to use: investment protection, effort to implement, performance, leverage, host independence, manageability, scalability, interoperability, extensibility, and portability. In different environments, each of the criteria will carry different weight.

Although the APPN architecture was designed to allow dependent LUs, there are three basic challenges. First, although APPN supports dependent LU data natively, it cannot support the SSCP-PU and SSCP-LU control sessions, so either these sessions or the information that flows on them must be handled. Second, dependent LU BINDs rely on subarea nodes for routing—APPN nodes need to be able to add routing information to dependent LU BINDs. Third, if the dependent LUs will be requesting access to applications on APPN, the application must support secondary LU initiate. ■

(continued from page 1)

Dependent LU Server/Requester Model

IBM's strategic approach for supporting dependent LUs over APPN is called the dependent LU server/requester (dLS/R) model. This model was first discussed as a statement of direction in March 1992 as "enhanced support for dependent LU routing." (*SNA Perspective* has chosen to use a lower-case letter for the abbreviations dLR and dLS so the latter will not be confused with the data link switching (DLS) support for SNA and NetBIOS traffic on the IBM 6611 router. IBM may change the data link switching abbreviation to DLSw.)

IBM has not officially announced dLR or dLS but has made some information available. Our discussion is based on this information from IBM and *SNA Perspective*'s analysis of how it will most likely operate.

Arbitrary Attachment to APPN

The dLS/R combination extends the dependent LU support provided in VTAM 4.1. The peripheral node would no longer need to be logically adjacent to its boundary function. It could be attached arbitrarily to an APPN network through any node with dLR support (see Figure 4).

The dLR implements a subset of the boundary function including, for example, session-level pacing. For the dependent LUs, dLR is the logical equivalent of the boundary function.

Where dLS/R Will Run

The dLS code will reside in an APPN network node, such as VTAM, which also has SSCP services such as directory support. The dLR code will reside in an APPN node, such as a 3174, which supports dependent LUs. Architecturally, dLR could be implemented in either a network node or an end node.

Each dLS can support more than one dLR. Each dLS can also support access to many application hosts.

A single dLR can support multiple PUs, including downstream PUs. Each PU 2 can support up to 255 dependent LUs, as they do today. These PUs must be logically adjacent to the dLR node.

Initially, dLR should be able to access a different dLS for each PU it supports, if needed. A backup dLS may be allowed.

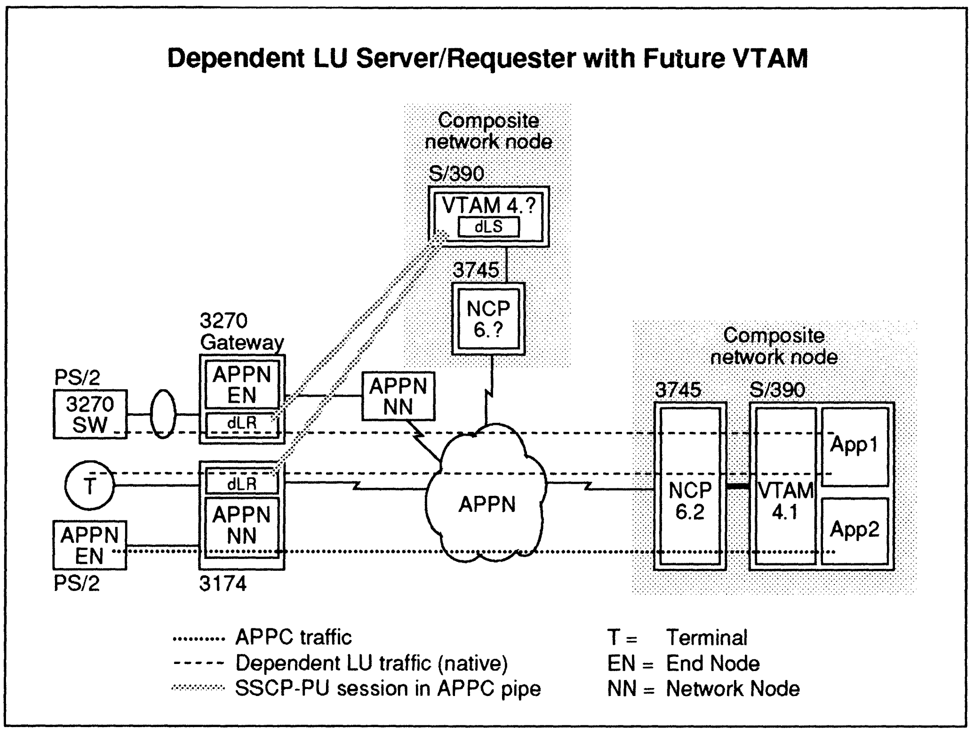


Figure 4

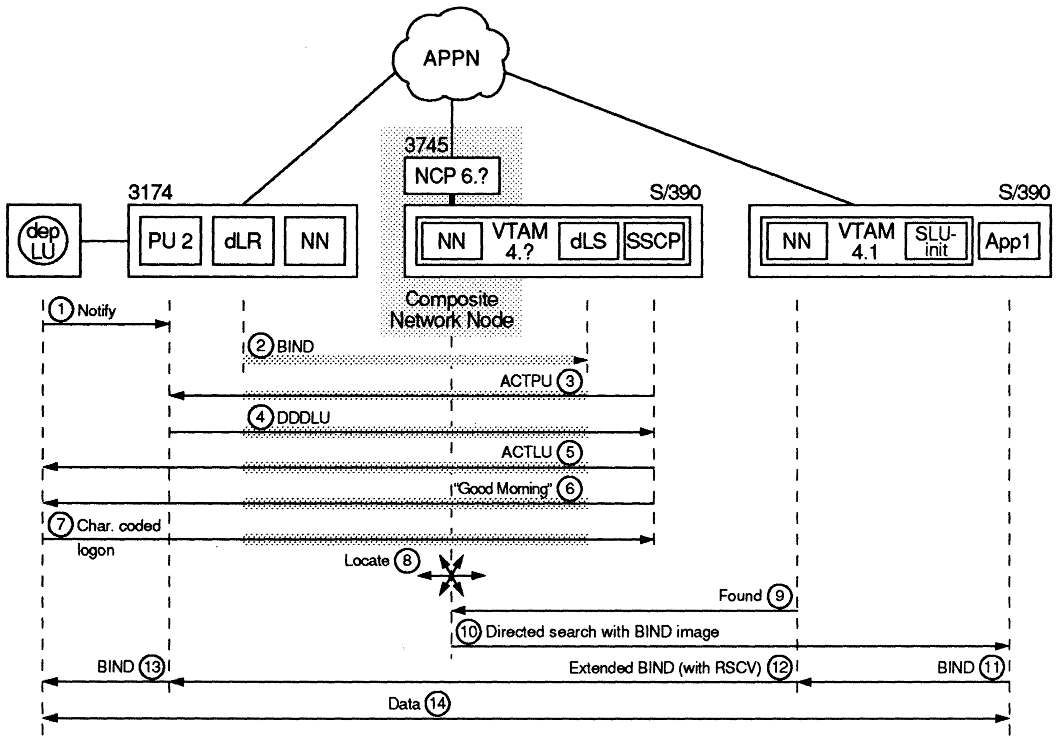
dLS/R Session Establishment

The purpose of dLR and dLS is to encapsulate the SSCP-PU and SSCP-LU sessions over APPN. The LU-LU data runs natively over APPN. A simplified diagram of dLS/R session establishment is shown in Figure 5 on page 10.

Upon powering on, the dependent LU issues a notify.

If not already established, an APPC session is then bound between a dLR and its dLS.

dLS/R Session Establishment



* Assume network nodes are known to each other through TDUs.
* Assume dLS is known to dLR (first release: system defined; second release: self-registration).

(1) Dependent LU powers on, sends a notify.
(2) dLR establishes an APPC session with dLS.
(3) VTAM sends ACTPU for SSCP-PU session; dLS encapsulates it.
(4) If LU is not predefined in SSCP, PU registers LU using dynamic definition of dependent LUs (DDDLUs); dLR encapsulates it.
(5) VTAM sends ACTLU for SSCP-LU session; dLS encapsulates it.
(6) VTAM sends a "Good Morning" message (menu of logon options) on SSCP-LU session; dLS encapsulates it.
(7) Dependent LU sends a character-coded logon specifying App1; dLR encapsulates it.
(8) If the application is not on the local host or database, dLS node does a broadcast search across APPN (and searches subarea).
(9) The network node serving the target application responds.
(10) dLS, through its network node via a directed search, requests the application to send a BIND to the dependent LU.
(11) Application generates a BIND and forwards it to its network node.
(12) The application's network node determines best route, appends RSCV to BIND, and sends. (Route need not go through dLS node.)
(13) dLR intercepts the BIND, removes the RSCV, and sends to LU.
(14) All LU-LU traffic is sent natively across this route.
All SSPU-PU flows, such as network management data, are sent over dLR/dLS session.
All SSCP-LU flows, such as Response Time Monitor, are sent over dLR/dLS session.

- In this example, the application node must support APPN secondary LU initiate. Under VTAM, APPN secondary LU initiate support was added in VTAM 4.1.

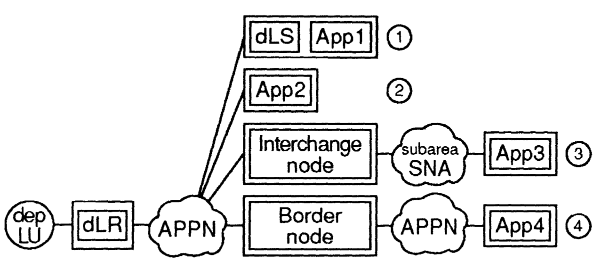
Figure 5

The traffic for session initiation from a dependent LU travels encapsulated in this APPC session between the dLR and the dLS. This single APPC session will support all the SSCP-PU and SSCP-LU sessions between the dLR and dLS pair. All functions and flows that run on the SSCP-PU or SSCP-LU session will still run on those sessions encapsulated between dLR and dLS.

If the SSCP-PU session is not active, the host sends an ACTPU. If the LU is not predefined in the dLS VTAM, the dLR node can dynamically register the LU in an NMVT through the dynamic definition of dependent logical units (DDDLU) support which was added to VTAM 3.4. The host will then send an ACTLU.

The usual "good morning" message is sent on the SSCP-LU session offering the LU its usual menu of options. The user makes a selection, which generates a character-coded logon message to the host. In response to the logon, the SSCP looks for the target application.

dLS/R Supports Access to Any Application Accessible through APPN or Subarea SNA



① Application on same node as dLS
② Application on separate APPN node
③ Application on subarea SNA network (VTAM 4.1 supports interchange node)
④ Application on a separate APPN network (A future release of VTAM will support border node)

- Unless all LU-LU sessions will be started by the application, the application's node must support secondary LU initiate.
  - For subarea SNA, secondary LU initiate has long been supported in VTAM.
  - For APPN, secondary LU initiate support is in VTAM 4.1.
- In a similar fashion, dLR could exist across a subarea SNA or separate APPN network from its dLS.

Figure 6

If the application is not on the same node as dLS, dLS performs a search across subarea SNA, APPN, or a mixed network using standard SNA and/or APPN search procedures.

The remainder of this dLS/R discussion assumes that the target application is also on APPN. However, the application could also be on the same node as the dLS. It could also be on a subarea SNA network or another APPN network, as shown in Figure 6.

With dLS/R, the target application node does not need dLS. However, if the application is on APPN and dependent LUs will be requesting sessions, the application's node must support APPN secondary LU initiate. (Secondary LU initiate is discussed in detail in part I.)

Our example assumes that the network node at the dLS does not know the location of the target application (and does not have access to a central directory server) and so must send a broadcast search. The network node supporting the target application replies to locate request.

The target application is instructed to send a BIND to the dependent LU. This is done in a directed search, which contains the name of the dependent LU's network node, the BIND image (supplied by the SSCP associated with dLS), and the secondary LU initiate parameter.

The target application generates a standard dependent LU BIND and passes it to its network node. The network node supporting the target application determines the optimal route to the dependent LU. It is important to note that the dLS node need not be in the session path.

The network node adds the routing information, or RSCV, to the BIND, which creates an extended BIND, and sends it.

The RSCV is removed by dLR and a standard dependent LU BIND is delivered to the dependent LU.

After the BIND is received and replied to, the dLS is informed by the application node that the requested LU-LU session has been established.

All LU-LU data for that session flow natively across APPN on that path. The specific dependent LU is identified by a local form session identifier (LFSID) in the header. The LFSID Address Space Assignment reserves a range of 255 addresses for dependent LU sessions in any node, from X'01',X'01' to X'01',X'FF'.

SSCP Sessions Remain Active

The SSCP-PU and SSCP-LU sessions remain active within the dLS/R APPC session. All the usual information that flows on these sessions continues unchanged. If this session goes down, it is recovered just as in NCP takeovers today—all session awareness is maintained. Further, the LU-LU session, which does not flow through this path, continues unimpeded by a failure of the dLS/R session.

Encapsulation of the SSCP traffic is done by way of an LU 6.2 general data stream (GDS) identifier which specifies SSCP data such as ACTPU, ACTLU, and character-coded logons. These control flows are actually encapsulated in two one-way APPC sessions.

VTAM Support for dLS

IBM stated in March that a future release of VTAM will support dLS and a future feature on the 3174 will support dLR.

The next VTAM release, VTAM 4.2, will probably be announced in mid-1993. *SNA Perspective* expects that dLS will be supported in VTAM 4.2. There will be fewer changes between VTAM 4.1 and 4.2 than between VTAM 3.4 and 4.1, so IBM could ship VTAM 4.2 as early as the end of 1993. However, a more realistic plan would be not to expect VTAM 4.2 and dLS/R until first or second quarter in 1994.

Still, readers are reminded that VTAM 4.1 supports dependent LUs (0, 1, 2, and 3) natively across APPN for all current configurations. What dLS/R adds is the ability for a dependent LU to connect arbitrarily to the APPN network.

3174 Support for dLR

We expect that dLR for the 3174 will first be released under the request for price quotation (RPQ) process, which allows IBM to offer new features without a new release of 3174 configuration support. Although the 3174 development staff is probably quite busy developing the Ethernet adapter and enhancing its TCP/IP support (which we believe customers will buy as soon as they are available), we expect that the 3174 dLR support will be available by the time dLS ships, probably in the first part of 1994.

IBM plans to implement dLR first in the 3174, which does not support end node. However, *SNA Perspective* believes that dLR would not necessarily need to be in a network node; the architecture allows it to also be implemented in an end node.

Other dLR Support

IBM has not discussed which other IBM nodes might receive dLR support. We believe that OS/2 Extended Services Communication Manager, AS/400, SNA Services/6000 on the RS/6000, 6611, and its own 3270 emulators would be likely candidates. After the 3174, the 6611 is the most likely candidate, since IBM made a statement of direction that the 6611 will support subarea SNA over APPN. We expect that dLR for OS/2 would be implemented next.

IBM has also not announced whether a future release of its licensed APPN network node code would contain dLR, or whether it would publish dLR as part of the APPN end node (if dLR can be implemented in an end node). IBM has said that adding dLR to the licensed network node code would be consistent with its APPN strategy, which sounds like a strong hint to us. We hope it is available as soon as possible so developers' products can be ready to ship when dLS is available.

SNA Perspective hopes that IBM will also publish dLR for implementation in an APPN end node. Many 3270 suppliers of 3270-compatible controllers, PC adapters and software, and protocol converters who are not looking for big business in the APPN network node market would be delighted to add more life into their new products and offer enhancements for their installed products. This could help create a market for dLS/R. If IBM does not do this, these vendors would be likely to put their energy instead into a third-party encapsulation approach.

VTAM-Lite for the AS/400?

The first release of dLS/R will support LU types 0, 1, 2, and 3. IBM has not stated when or whether it will implement dLS on the AS/400 to support LU types 4 and 7. *SNA Perspective* believes that it is unlikely to do so since dLS seems to require many VTAM functions and many AS/400 networks exist without a VTAM node. The IBM architecture team could extract a set of only those SSCP functions necessary for dLS and for APPN secondary LU initiate, a kind of VTAM-lite, which might be more palatable to the AS/400 community. Since the AS/400 already has an encapsulation approach for LU 4 and 7 and porting dLS in its current form seems unwieldy, it seems unlikely to us that IBM's Application Business Systems line of business will make the effort in the foreseeable future.

The SSCP Factor

If not predefined in VTAM, dependent LUs can be registered dynamically today over subarea SNA by their peripheral node across the SSCP-PU session using dynamic definition of dependent LUs (DDDLUs). DDDLU is implemented in VTAM 3.4 and future releases.

The dLS needs to have the dLR in its system definition if the sessions will be initiated by the application, such as autologons. If the sessions will be initiated by the dependent LU, the dLR needs to know about its dLS. In a future release, *SNA Perspective* expects the dLR and dLS will be able to dynamically find each other.

Figure 7 shows the dLS/R model discussed in IBM's statement of direction and *SNA Perspective*'s expectation of future product implementations.

Benefits of dLS/R

There are several advantages to dLS/R. First, the LU-LU session is not affected if the dLS/R session goes down and the SSCP sessions are recovered when the dLS/R session is recovered. Second, users get the full benefits of APPN, since dLS/R can let the application's network node server calculate the best route directly to the dependent LU. Third, all management, accounting, and security applications will work. The traffic that runs on the SSCP control sessions can run unmodified. Fourth, since the data is running natively, there is no overhead or performance effect from encapsulation. Fifth, since the dependent LU data runs natively over APPN, it maintains visibility for network management all the way to the dependent LU connectivity. Finally, the dLS/R approach will more efficiently scale up for large networks with many nodes needing access to multiple locations.

Disadvantages of dLS/R

On the other hand, dLR and dLS require three pieces of code—dLR support at the dependent LU, dLS support in the server host, and APPN secondary LU initiate support (if the environment needs it) in the target application node.

As with any proprietary solution, there is a perceived risk to customers and other vendors in IBM owning dLS/R, even if it publishes or licenses dLR.

Additional advantages and disadvantages of dLS/R compared to an encapsulation approach are discussed below.

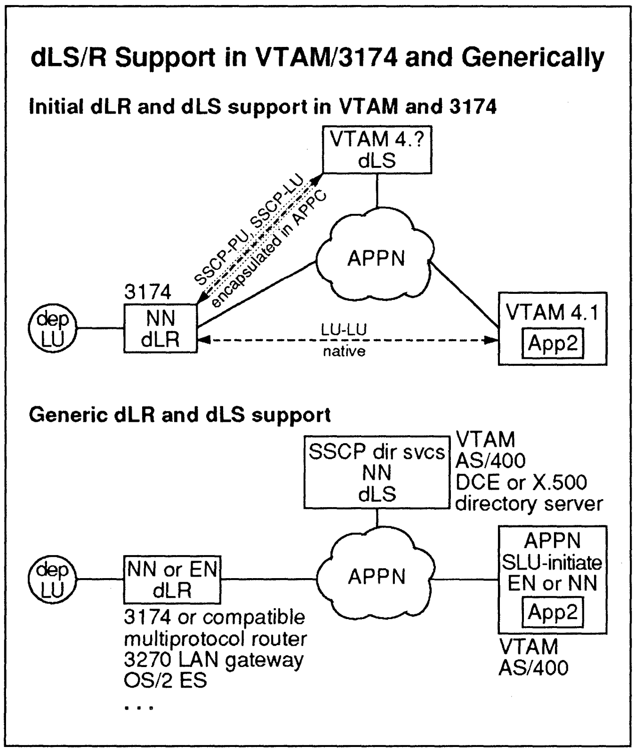


Figure 7

Encapsulation

An alternative to dLS/R is encapsulation. IBM's strategic approach to support dependent LUs over APPN is dLS/R, but the company would be interested in someone else developing an encapsulation product (see *Architect's Corner* in *SNA Perspective*, October 1992).

The industry is discussing several very different approaches to encapsulation. *SNA Perspective* has identified the three main options. These are listed below and shown in Figure 8.

- Dependent LU tunneling—SSCP control sessions and dependent LU-LU sessions encapsulated inside of APPC

To be competitive with dLS/R, an encapsulation scheme should be openly developed—for example, by a university or a forum of vendors and users—and the various vendors' implementations should be compatible.

To provide flexible resource access, an encapsulation solution should support search capabilities equivalent to SSCP directory services to find the target application across the APPN network. Otherwise, all target applications will have to be preconfigured in the encapsulation client.

This encapsulation code should be relatively small and able to run in several environments. Also, all dependent LU types should be supported using a similar structure.

Supporting the data stream is not a problem—APPC was designed with special GDS identifiers for carrying the 3270 data stream and several other data streams that are usually supported by dependent LUs. The challenge will be in mapping or supporting the control information which usually flows over the SSCP-PU or SSCP-LU sessions or as special request/response units (RUs) that flow on the LU-LU session.

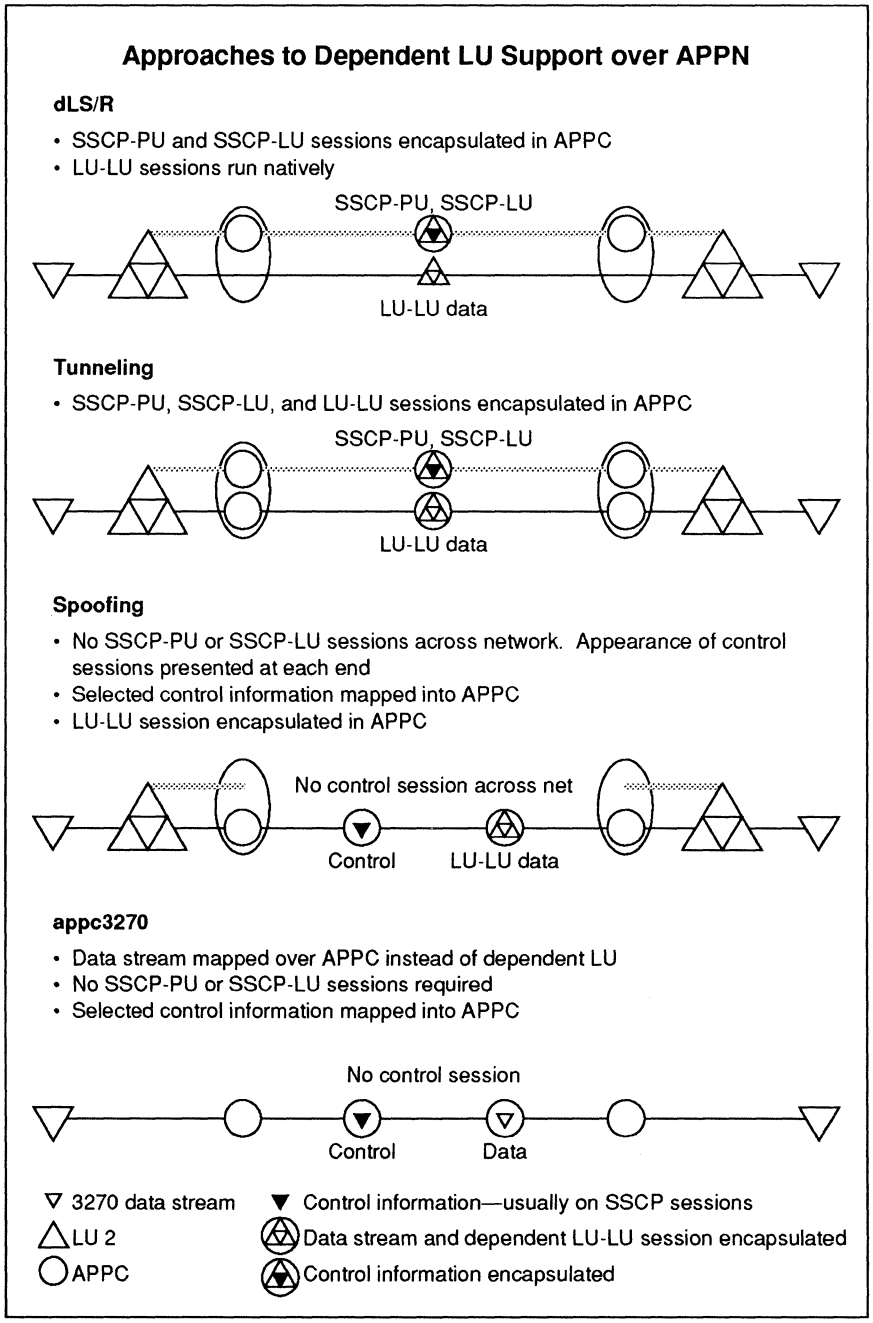


Figure 8

- Dependent LU spoofing—LU-LU data encapsulated in APPC; subarea PU and dependent LU appearance at either end but no SSCP control sessions across APPN; selected control flows mapped into APPC
- appc3270—data stream intercepted and mapped over APPC instead of dependent LU; similar to tn3270

Dependent LU Tunneling

Dependent LU-LU tunneling is the simplest approach. It would involve encapsulating in APPC not just the SSCP control sessions, as dLS/R does, but the LU-LU session as well. Encapsulation and decapsulation would be done at each end of the tunnel. This is similar to the AS/400 5494 Remote Attach feature, which encapsulates the LU 4 and 7 sessions in APPC between the controller and the AS/400. The LU-LU data would probably run on a separate APPC session from the control flows.

Dependent LU Spoofing

Another approach is for the peripheral node (at the workstation or client side) to act as a proxy host, spoofing a PU 4 appearance. For example, this proxy would locally emulate the activate PU (ACTPU) and activate LU (ACTLU), though no actual SSCP-PU and SSCP-LU sessions would exist across the network. On the host side, the server would act as a proxy workstation, presenting a PU 2 appearance. For example, it would accept the actual ACTPUs and ACTLUs from the host by emulating the workstation PU and LUs.

Much of the information that flows on the control sessions is not required if the session is not actually running on a subarea network. However, some information that usually runs on the control sessions—such as network management alerts or character-coded logoffs—would need to be supported. These flows could be mapped to run on an APPC session between the encapsulators at each end.

As with tunneling, the LU-LU session would be encapsulated in an APPC session across the APPN network.

appc3270

The appc3270 approach would be like tn3270, which sends the 3270 data stream in a telnet session over TCP/IP.

In tn3270, the tn3270 client at the workstation and the tn3270 server in or near the host communicate through a telnet session over TCP/IP. The tn3270 server includes a fake LU to spoof the host application to think it has a type 2 LU-LU session. No SNA runs across TCP/IP—the 3270 data stream is mapped to telnet instead of to LU 2.

Similarly, instead of supporting dependent LUs, appc3270 would replace them. Instead, it supports the data stream natively on APPC. In this, it differs significantly from dLS/R and the other two encapsulation approaches.

The appc3270 approach violates one of the preferred attributes of a dependent LU support solution because it requires a change in the workstation software and, depending on the implementation, in the application.

However, because of industry interest and the popularity of tn3270, *SNA Perspective* believes this approach could be important.

Benefits of Encapsulation

A major benefit of encapsulation is that it works with existing VTAMs, existing 3174s, existing emulators, and so on. Users may continue at lower release levels, especially for systems they do not intend to migrate. Users would not need to upgrade an existing 3174 to APPN, for example, which can cost about $10,000, to add dLR. Instead, the 3174 could be connected to an encapsulator on an inexpensive OS/2 platform. Further, users could access applications on hosts with any release level of VTAM; the host would not need native APPN support.

Second, encapsulation would only need two new pieces of software, while dLS/R uses three—dLS, dLR, and, usually, APPN secondary LU initiate at the target application. Rather than a single server node for several hosts, though, server software would be used at each application node.

Third, if dependent LUs can be successfully supported on APPN over APPC, other types of traffic, such as sockets or asynchronous traffic, could be similarly encapsulated in or mapped to the same APPC transport. Each protocol would involve a separate and equally difficult process; there is no leverage here. However, consequently, any enhancement to APPC would equally benefit all the encapsulated protocols.

Finally, encapsulation can take advantage of the more flexible session initiation and network addressing offered by independent LUs.

Disadvantages of Encapsulation

Without some SSCP-like directory services support, all target applications across APPN must be preconfigured at the workstation end. A resource location service could be provided somewhere on the APPN network. In the long run, a centralized directory server based on the Open Software Foundation (OSF) Distributed Computing Environment (DCE) or on X.500 directory services could be adapted provide this service over APPN and SNA, but no products are available yet to which this service could be added. This server would need to be able to search across subarea SNA, APPN, and interconnected networks, which is one of the main functions of dLS and its interaction with VTAM.

Another alternative would be a new protocol at the peripheral end that could discover the contents of the SSCP database. Another option would be to require the end user to specify a fully qualified LU name—but this violates the goal of not changing existing systems or procedures at either end.

Another concern is raised about network management, which is relevant when any traffic is encapsulated. NetView cannot see inside encapsulated traffic, including the encapsulated control sessions with dLS/R, which makes problem determination more difficult. Also, encapsulation approaches that do not fully support the SSCP control sessions may not support all the expected network management capabilities.

Since encapsulation adds an additional step for each packet over native transport, issues of overhead and performance are raised.

Further, with appc3270, it will be a challenging process to map all the 3270 control functions (those that usually run on the SSCP-PU and SSCP-LU sessions). Even tn3270, after many years on the market, does not support all the features that today's 3270 emulators provide. The 3270 working group of the Internet Engineering Task Force is struggling to map additional support to tn3270.

Finally, an encapsulation solution may support many but not all types of existing devices. It might be more flexible but more limited—a common design decision.

Encapsulation and dLS/R Compared

Both approaches address a common, significant need—supporting dependent LUs across APPN. This section discusses ten criteria relevant in the selection of encapsulation or dLS/R.

As discussed earlier, we do not expect dLS/R to ship until some time in 1994. Since the development of an open encapsulation approach has not yet begun, products based on this approach are unlikely to appear before that time either.

However, their absence need not inhibit a user from beginning to incorporate APPN in a subarea SNA network because, as discussed in part I, VTAM 4.1 provides significant dependent LU support.

Investment Protection

A primary goal of either approach is to require as little change as possible at either end of the network. This includes installed host applications and application subsystems, application definitions to the operating and communications systems, user interface, terminals, workstations, controllers, gateways, adapters, cabling, and other supporting hardware and software. It should also require no retraining for end users.

The dLS/R approach requires no changes to an existing environment, and also allows moves and changes in dependent LU workstations and nodes without host changes. It is unclear how the encapsulation approach will work in this regard or where the client and server code will reside. The appc3270 approach would require more changes than the other two techniques.

The dLS/R will probably require the least amount of changes in existing environments. The greatest change would be needed for appc3270.

Implementation Effort and Cost

Each approach must be considered with regard to the size of the effort involved in its implementation.

Since these products are not announced, no pricing is available. The dLR and dLS capabilities will probably be included at minimal or no charge with a future release of 3174 Configuration Support and VTAM. However, for dLS/R, the user needs to upgrade to the latest VTAM and 3174 code and the target application node will probably need APPN secondary LU initiate (VTAM 4.1). In addition, to handle dLS/R, many APPN nodes already installed will need to be upgraded because they did not implement the architectural support for dependent LUs.

An encapsulation technique may be less expensive to implement overall because it will probably work with lower level releases, even with pre-LEN node VTAM. Therefore, the cost and effort of upgrading is not required, which is helpful for systems the user does not intend to migrate. Also, since it will not require VTAM, as dLS does, it could be popular in sites without a mainframe, such as AS/400 networks, or where the mainframe is overloaded.

However, encapsulation will probably require more manual definition at the client side. Also, an encapsulation server will be needed at every application node, while dLS is only needed on one node.

With either approach, the implementing vendors will probably need to pay APPN end node patent license fees or APPN network node license fees and royalties for the APPN code with which the dLR or encapsulation software will interact.

When balanced out, we expect that the end-user implementation effort between the two approaches will probably be equivalent. The encapsulation technique will probably need more ongoing manual adjustments while the dLS/R approach will be automatically supported by regular network maintenance. The initial cost for dLS/R, taking system upgrades into account, may be higher.

Performance

Performance considerations should take into account overhead and throughput and also where the overhead hit is taken (on the mainframe or an offload box).

Logic indicates that, since encapsulation and decapsulation add two steps to processing each frame, the encapsulation approach will use more overhead and impact throughput. The size of this impact cannot be known at this time. IBM has said, though, that experience with encapsulation on the AS/400 indicates that the amount of overhead is minimal and the percent degradation is unnoticeable. We do not expect this would be a significant concern in moderate-sized installations at current network speeds, but could be more problematic in larger networks or with emerging very high speed networks.

If dLR and dLS code becomes too ambitious and large, this may outweigh its performance advantages. The encapsulation code on the 5494 Remote Attach, discussed in part I, takes only about 2 KB of memory, though it is less flexible than dLS/R.

With some encapsulation approaches, a fixed pipe may be necessary between each dependent LU node and application. With other encapsulation approaches, two APPN routes may be necessary, one from the device to a server and another from the server to the target application. With dLS, the LU-LU traffic need not go through the dLS node; the route is calculated directly to the dependent LU.

Leverage

Leverage is the ability to reap multiple benefits from any additional investment. The amount of leverage users will experience with either approach will depend on the types of traffic on their network.

The initial release of dLS/R will support LU types 0, 1, 2, and 3, while appc3270, for example, will probably only support LU 2 for 3270 displays and perhaps LU 3 for 3270 terminals. An investment in dLS/R will thus support a wider range of products.

On the other hand, once the encapsulation technique is developed, it could be adapted to support other types of traffic over APPN networks, such as OSI, BSC, asynchronous, and TCP/IP. IBM has made a statement of direction, for example, to support sockets over APPC and APPN. Each protocol would involve a separate and equally difficult process; there is no leverage here.

Consequently, any enhancement to APPC would equally benefit all the protocols. With dLS/R, the LU-LU traffic carried natively would not benefit from enhancements to APPC.

Host Independence

The criteria of host independence has two aspects: whether it is possible without VTAM (such as an AS/400 network) or possible with an application or controller on a system with a noncurrent release of VTAM.

The AS/400 community would prefer a solution that does not require VTAM, as dLS currently does, and so IBM is unlikely to implement the dLS/R scheme for LU types 4 and 7, especially since it already has an encapsulation technique for them.

The dLS code will require the then-current release of VTAM (probably VTAM 4.2) for the server node as well as probably APPN secondary LU initiate in the target application nodes (VTAM 4.1). This may not be a problem in many networks since users who are moving to APPN are those who are upgrading their mainframes anyway.

The various encapsulation approaches could offer a variety of levels of host independence. As discussed above under "Implementation Effort and Cost," encapsulation could be designed to work with any release of VTAM or PU 2 code and would not require them to have APPN.

The encapsulation technique will probably offer greater host independence.

Manageability

An important element of any networking decision is the ability to manage it. For dependent LUs, this means the ability of NetView to view and access them and their sessions and for VTAM to issue network management commands.

With dLS/R, all existing network management support will work. For example, the SSCP-PU session can carry alerts and the SSCP-LU session can support Response Time Monitor. Further, since the dependent LU data is carried natively across APPN, it is visible to NetView.

Depending how the encapsulation approach is done, some of the network management services that use the control sessions may be unavailable. Further, the encapsulated data is hidden from NetView, including the control sessions encapsulated in dLS/R. Thus, some of the ability to detect and analyze problems or inefficiencies is lost. If the session fails, for example, this fact can be reported to the focal point, but further information on the reason for failure may not be available.

Both approaches will suffer from NetView's limited support for APPN networks at this time, such as its inability to track APPN network changes.

With regard to network management, the dLS/R approach appears to offer greater advantages.

Scalability

Scalability is the ability of a solution to be able to handle increasingly large networks while maintaining its performance and flexibility.

The dLS/R approach will scale up more gracefully. Until an alternative directory server is available, the encapsulation approach will probably be less efficient in larger networks.

Interoperability

Interoperability in today's market is primarily related to how each approach can work with or next to TCP/IP. Both approaches will probably find it equally comfortable (or uncomfortable) associating with TCP/IP.

The tunneling and spoofing encapsulation techniques are similar in concept to how multiprotocol routers are supporting SNA. However, these techniques will probably not support any interoperability with routers.

It is possible that tn3270 and appc3270 could interoperate with a gateway. Similarly, dLS/R is not necessarily limited to APPN. Technically, dLS/R could be mapped to support dependent LUs over TCP/IP.

Extensibility

Extensibility involves how easily can it be adapted to support new capabilities. It is difficult to predict which approach will be more easily extensible, since the shape of future advances is unknown.

Both approaches will, in theory at least, be equally adaptable (or unadaptable) to support newer subnetwork technologies such as ATM, SMDS, and frame relay. However, as the network speed increases, the performance impact of encapsulation will be magnified.

The dLS/R scheme is probably limited to supporting SNA dependent LUs (though being limited to a market of only ten million or so installed devices is a "problem" many vendors would like to face). Its first release will support LU 0, 1, 2, and 3. It may not be implemented to support LU 4 and 7 for a long time.

The encapsulation approaches will likely handle one or two LU types at first, probably LU 2 and 3. Since one encapsulation benefit is supposed to be a small, inexpensive solution, there may be separate products for different LU types.

Portability

Portability is the ease with which the code can be ported to a range of platforms. *SNA Perspective* expects that IBM will make dLR code available by either publishing or licensing it. Other vendors could then implement it along with APPN in their controllers, gateways, and routers.

However, IBM will own the server side—dLS currently depends on VTAM and is unlikely to run on other platforms in the foreseeable future. It could be implemented on an AS/400, but the AS/400 group seems to have chosen the encapsulation approach. In the long run, it could be adapted to run on a DCE or X.500 directory server.

As with any proprietary solution, there is some risk to customers and other vendors in IBM owning dLS/R, even if it publishes or licenses dLR. Even if IBM is operating with the best of intentions, it will develop enhancements for dLS/R that best serve the largest percentage of its preferred client base, which does not include all users or other vendors.

The code for the encapsulation approaches we have examined will probably be developed in the public domain and so will likely be designed with portability in mind.

SNA Perspective expects that the encapsulation client code may be more easily ported than dLR code.

Summary

User Requirements

In order for APPN to be a successful successor to subarea SNA, it must support the millions of existing dependent LUs. This support can be either native or though encapsulation.

In addition to providing APPN routing for dependent LU data, an ideal solution should have four additional qualities.

- Support existing user interfaces, existing devices—terminals, PCs, controllers, gateways, etc.—and existing applications transparently, that is, without any changes to them beyond a basic APPN connection.
- Support all data and control information, whether it usually flows over the SSCP-PU, SSCP-LU, or LU-LU session.
- Support SSCP or SSCP-equivalent directory services for resolution of local names to LU names and for searching and access across APPN, subarea SNA, and mixed networks.
- Provide as many as possible of the benefits of APPN to the dependent LUs, including optimal route selection and host-independence.

Future Support

The dLS/R model is IBM's strategic approach. A future release of VTAM will support dLS and a future release of 3174 code will support dLR. This model eliminates the adjacent boundary function requirement in VTAM 4.1. Either the dependent LU can connect to an APPN network at any node with dLR or dLR can be added to the dependent LU node itself. *SNA Perspective* expects that this future release of VTAM will not ship until mid-1994.

IBM and several third parties are also discussing the development of encapsulated support for dependent LUs over APPN. This could be done in several ways—tunneling, spoofing, or appc3270. Since these development efforts are in their early stages, *SNA Perspective* does not expect any solutions to be available until the end of 1993.

Encapsulation versus dLS/R

Each approach has its advantages and disadvantages for different environments. Users will need to carefully consider their environments to make a choice.

The dLS/R approach will scale up more gracefully to larger networks, mixed subarea/APPN networks, and very high speed networks. It will provide for more flexible and dynamic configurations for dependent LUs. It also provides full SSCP support. On the other hand, it is currently VTAM dependent, requiring a future VTAM release, probably VTAM 4.2, at the dLS node as well as VTAM 4.1 at most application nodes. IBM owns and controls dLS/R, which makes some customers and other vendors uncomfortable. But *SNA Perspective* hopes that IBM will publish or at least license dLR so that vendors of controllers, multiprotocol routers, and gateways can access dLS.

An encapsulation approach will probably be less expensive, in part because it will not require upgrades to current revisions. It will serve best for small- to moderate-sized APPN networks that have only a few LU types. *SNA Perspective* expects that the first encapsulation products will probably only support LU types 2 and 3, although the AS/400 already offers an encapsulation approach for LU types 4 and 7. Further, in the likely absence of access to SSCP directory services, all the target applications must be preconfigured in each encapsulator for all the devices it supports. Ideally, vendors and users will find or create a forum to support the open development of one encapsulation approach as an industry standard. It is more likely that several of the techniques we discussed will be offered by different vendors so there will be some eventual shakeout in the market and, thus, a risk for users in investing in early products.

Planning Implications

SNA Perspective believes users can feel comfortable continuing to make investments in dependent LU products and systems today, knowing that they will be supported if a move is made to APPN.

Although the products discussed in part II are not available today, an understanding of the types and time frames for dependent LU support on APPN will assist users in their migration and integration planning. Since early 1991, we have watched the emergence of support for subarea SNA traffic over TCP/IP. More recently, we have been watching whether, how, and when IBM and other vendors will support APPN. Users can and should begin now to discuss with these vendors the approach they would prefer for supporting their dependent LUs over APPN. ■

LU Review

A logical unit (LU) is a port through which an end user (either an application or an I/O device such as a terminal) accesses an SNA network. An LU-LU session is set up between two applications or a device and an application and data flows between them over that session. The LU is implemented at approximately layers 4–6 of the OSI reference model. It provides services to end-user programs and devices and interacts with the path control network at the lower layers.

There are several LU types. LU 2 is the most prevalent SNA LU session type in use. LU 2 supports 3270 diplays and LU 1 and 3 support 3270 and SCS printers. LU 7 supports 5250 terminals for the AS/400 and S/3x and LU 4 supports printers for the same systems. LU 0 is a special LU for which SNA layer 6 does not define a standard set of services—the user defines these services and functions. LU 0 is commonly used for automated teller systems and banking terminals in financial companies and for point-of-sale systems in retail environments.

LU 6.2, also called Advanced Program-to-Program Communication (APPC), is the most recent LU type (first announced in 1982). It is a converged LU and is designed to support communication between applications and intelligent devices without the host support required for the other SNA LU types.

Note: Technically, LU 6.2 refers to the protocol services in layers 4–6, while APPC was defined as the application programming interface (API) to LU 6.2. However, IBM has recently made a marketing redefinition so that the term APPC is now identical to LU 6.2 and an APPC interface in any implementation is now called the "native APPC API." Since we believe that the internetworking community can relate more easily to the term APPC than the LU terminology and since SNA, in order to thrive, must coexist with the internetworking community, *SNA Perspective* has chosen to adopt this redefinition and will use the terms LU 6.2 and APPC interchangeably.

Physical Unit

The physical unit (PU) is the software component that manages and controls all the resources in an SNA node. It is implemented at about layers 3 and 4 of the OSI model. Each LU is associated with and supported by a PU either in the same or an adjacent device. Traditional subarea SNA PUs are implemented in a hierarchical fashion.

The PU in a host is called PU 5 and is implemented in the System Services Control Point (SSCP) within VTAM. The SSCP provides control, coordination, and directory services for all the PUs and LUs it supports on the network. The SSCP also supports all the application LUs on the host.

The subarea PU in a communication controller, such as the 3745, is called PU 4. A PU 4 does not usually support LUs, except for virtual LUs to support X.25 communication and non-SNA terminals. A communication controller performs subarea SNA routing and may support peripheral nodes. This peripheral node support is called the boundary function.

There has never been a PU 3.

PU 2 is known as a peripheral node and is implemented in 3270 controllers, gateways, and protocol converters.

PU 1 is used for 5250 controllers. It was also used for an obsolete 3270 integrated terminal/ controller.

A newer PU type, PU 2.1, supports Advanced Peer-to-Peer Networking (APPN). This PU type primarily supports LU 6.2, but can support other LUs. To emphasize the independence of the software from the physical device, IBM has renamed PU 2.1 to be node type 2.1. When APPN support is added to an existing subarea PU, it becomes a type 2.1 node in addition.

The control point (CP) provides some of the services that SSCP provides in a subarea SNA network. Older type 2.1 nodes, called low-entry networking (LEN) nodes, did not include a control point, but the newer APPN network node (NN) and end node (EN) each contain a control point.

Host Requirements

For LU types 0, 1, 2, and 3, control sessions are needed between the host SSCP and the peripheral PU (SSCP-PU) and between the host SSCP and each LU (SSCP-LU). For LU 4 and 7, the AS/400 does not require an SSCP-PU session to the controller.

LU Review (continued)

Independent and Dependent LUs
An independent LU is one that requires no SSCP-LU session. Only type 2.1 nodes can support independent LUs. (Type 2.1 nodes can also support dependent LUs.)

Though LU 6.2 has been the primary independent LU implemented by IBM and others, the architecture allows an independent LU to use any LU type.

LU 6.2 is usually implemented as an independent LU. However, an LU 6.2 on a subarea PU or an LU 6.2 defined as a dependent LU on a type 2.1 node still require an SSCP-LU session.

Boundary Function
Boundary function (BF) is a PU 4 function that provides support for its adjacent peripheral nodes. BF is usually provided through a communication controller, but may reside in the host for locally attached nodes such as channel-attached 3174s. It transforms network addresses to local addresses, provides session sequence numbering for PU 1 nodes, and performs session-level pacing between the BF and the secondary LU in a peripheral node.

In subarea SNA, a peripheral node must be logically adjacent to its BF node. Logically adjacent means that its traffic cannot go through another SNA node on the way to its BF. (A 3270 controller with gateway support for downstream PUs is an exception because it supports the traffic transparently, acting as a sort of bridge.)

Distinction between Data Stream and LU
It is important, in considering different encapsulation approaches, to remember the difference between a data stream and an LU, though they are often closely related. For example, the 3270 data stream controls the screen while the LU provides the session that conveys the data stream.

Although the 3270 data stream is integrated closely with and is predominantly carried on LU 2, it can be implemented elsewhere. For example, tn3270 is a public-domain protocol that implements 3270 data stream over a telnet session on a TCP/IP network. ■

Architect's Corner

SNA Compass

by Dr. John R. Pickens

Technologists are drawing many "maps" to help navigation in the SNA environment—TCP tunneling, data link switching, Cisco's Advanced Peer to Peer Internetworking (APPI), collapsed source routing, source route transparent (SRT) bridging, SDLC passthrough, PU spoofing, etc. Many of these mapmakers are driven by the desire to differentiate, or to dominate, or to innovate, or to interoperate over, under, across, and through the installed (old) internetworking (WAN) base. No wonder users are greatly confused by the array of choices.

What is needed is not more maps, but a compass. A compass with a true north that can always be used as a point of reference. A compass that can be used on water or land, in the air or underground, whether traveling by road, tunnel, chunnel, or flyway. (Or whether by bridges, routers, links, LANs, MANs, or cloud technologies…)

Problems with Maps
Most SNA maps are needlessly complex and they often obscure the objective of reaching the final destination—interoperable multiprotocol (including SNA) bridging/routing. Consider the following sampling of SNA maps that are being proposed today.

- A vendor offers an IP-over-SNA tunneling solution. But the solution only provides intermediate system-to-intermediate system (IS-IS) compatibility with routers running RIP, a routing protocol with known operational limitations (said kindly).
- A vendor offers a "collapsed" source routing scheme, allowing users to create WANs with larger diameters (more hops). But more hops

increase the likelihood of hitting the latency "wall," introducing delays that cause unnecessary timeouts and retransmissions at the data link layer, user frustration, and even lost connections. Also, network management visibility is reduced.

- A vendor offers an SDLC tunneling service for subarea node (PU 4 properties), with SDLC transmission groups tunneled over TCP/IP. Tunneling introduces new delay characteristics that cause problems for subarea node software, especially in keeping multiple links synchronized.
- A vendor proposes a "better" APPN solution. APPN directory services spoofing. Tunneling over TCP/IP connections. Mapping APPN-based routing metrics to IP-based routing metrics. But what the user wants is just APPN.

Unfortunately, what one vendor does others often emulate, creating a downward spiral of complexity and confusion. Vendors seem prone to copy (or one-up) each other's "maps."

The current plethora of vendor maps has created a crisis of complexity. Users do not really want this complexity. In the words of a user recently quoted in *SNA Perspective,* "as far as possible we can't allow any end network to have any influence on the WAN." Or, paraphrased, "just give me a WAN that works."

Users want a WAN that is like the phone network. It works. It is transparent. It is simple. And it meets requirements.

True north for SNA can be described simply. Provide networking services, in transparent fashion, that are simple to install, maintain, and operate, and that are compatible (interoperable) with SNA end systems and intermediate systems. Or, stated a little less simply, provide bridging and routing services for SNA end systems and intermediate systems based on standards. Avoid complexity.

Both users and vendors have a role to play here.

SNA Compass—Vendors

Here are a few suggestions for vendors:

- Implement standards
- Tunnel, but sparingly

Where standards exist, implement them. Help migrate users to standards. Where standards do not exist, work with the relevant standards forum to create standards. Two examples are relevant to the SNA environment:

- IEEE 802.1D SRT bridging. SRT solves the topology/internetworking problem of token ring source routed and transparent LAN/WAN domains. The next step beyond 802.1D is 802.1G, which extends the standard to remote bridging. The standard-bearer here is IEEE 802.
- SNA routing. APPN is the (de facto) standard for SNA routing. In the short term, use APPN to route APPC sessions and bridge all else. In the long term, use the dependent LU requester extension of APPN to route all SNA session traffic. The standard-bearer is IBM. However, IBM needs to improve its ability to work with vendors on APPN extensions.

Much of what the multiprotocol vendors are offering today is based on tunneling. In the SNA environment, what is being tunneled is layer 2 (SNA data link layer) traffic. Many variations on the theme are offered—distributed name-to-address resolution schemes, local termination, passthrough, conversion, etc.

Data link tunneling has a place. Between APPN routers, for example, tunneling is a useful way to cross TCP/IP-only domains—like X.25 QLLC in which X.25 sessions are used to provide logical switched links between routers.

But, overall, I fear too much emphasis is being placed on tunneling, as though it is the endgame, the destination, or even the compass.

Tunneling has migration value for users who have not yet entered the multiprotocol routing domain, such as those with only OSPF or only SNA on the backbone. But, while it simplifies the backbone routing protocol, tunneling introduces complexities of its own—concatenated connections (LLC2-to-TCP-to-LLC2), lack of standards, latency (timeout) challenges, and additional administrative overhead. With care, such complexities can be managed, but they should be utilized sparingly, and only when standards-based IEEE 802 bridging and/or IBM APPN routing cannot be justified.

SNA Compass—Users

Users have a role to play also.

Insist on solutions that make sense and migrate easily. Understand the SNA compass, and insist on solutions that are consistent with its direction.

Acquire new systems based on standards or with plans to become compatible with the standards. For the short-term minded, this is the most difficult recommendation to follow. If simplicity and standards are agreed-upon objectives, then a commitment must be made toward migration. For the installed base, this is a tough commitment. Older systems—those that are not simple or are based on nonstandards or have outlived their value—have to be replaced.

Two examples exist today (and more, undoubtedly, will tomorrow) of systems for which migration should be planned for SNA traffic:

- Nonstandard (or earlier standard) bridges
- Non-APPN routers

A plethora of nonstandard bridges exist in the token ring environment. These bridges are not compatible with the current IEEE 802.1D SRT bridging standard. Remote bridges are also not compatible with the emerging IEEE 802.1G remote bridging standard (of course, the standard does not yet exist).

Rather than increasing the complexity equation through ad hoc gateways, plans should be made for replacement with standards.

Note: Users may have to migrate some end systems (software) also. A few end systems do not yet participate in the new procedures required of SRT standard bridging. In particular, end systems attached to source routing–capable media must be able to operate in both transparent and source routing modes.

With regard to "routing" SNA today, each vendor offers a tunneling solution and none of the solutions are common. Regardless of whether such a tunneling scheme might become standard, such as IBM's data link switching or Cisco's APPI, plans should be made to migrate toward the standard SNA routing solution—APPN and its enhancement, dependent LU server/requester.

The SNA Compass—Looking Forward

The SNA compass is pointing toward standard IEEE 802 bridging and standard APPN routing. The sooner vendors and users incorporate this compass into evaluating the plethora of networking road maps, the better off they will be.

One final comment should be made about APPN evolution. Beyond current APPN, the future is not easy to predict. As high-speed data link layer technologies catch on (fast packet, ATM, etc.), APPN+ will take on more of a control and bandwidth management flavor. If these technologies are delayed, then an integrated routing scheme—such as triple OSI-IP-APPN IS-IS—might appear as an alternative.

Whichever long-range APPN track is taken, many vendor maps will likely get drawn (and argued over) along the way. Only the compass indicating true north will point the way for both vendors and users. ■

• 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 • Marketing/Development: Alisson Walsh • Circulation Coordinator: Cheryl Roberts • Contributors: Dr. John R. Pickens • Typesetting and illustration: Aaron Lyon at dSIGN • The information and opinions within are based on the best information available, but completeness and accuracy cannot be guaranteed.