Systems Network Architecture (SNA) has evolved from an architecture that supported implementation of tree networks rooted in a System/370 to an architecture that supports multiple-system networks with capabilities such as alternate paths and parallel links. This paper describes the major SNA enhancements that have been implemented for multiple-system networks. As network configurations have become more complex, the problems associated with network growth, change, failures, recovery, and flow control have required solutions that permit continuous network operation. The SNA enhancements that address these problems are also discussed.

SNA multiple-system networking

by J. P. Gray and T. B. McNeill

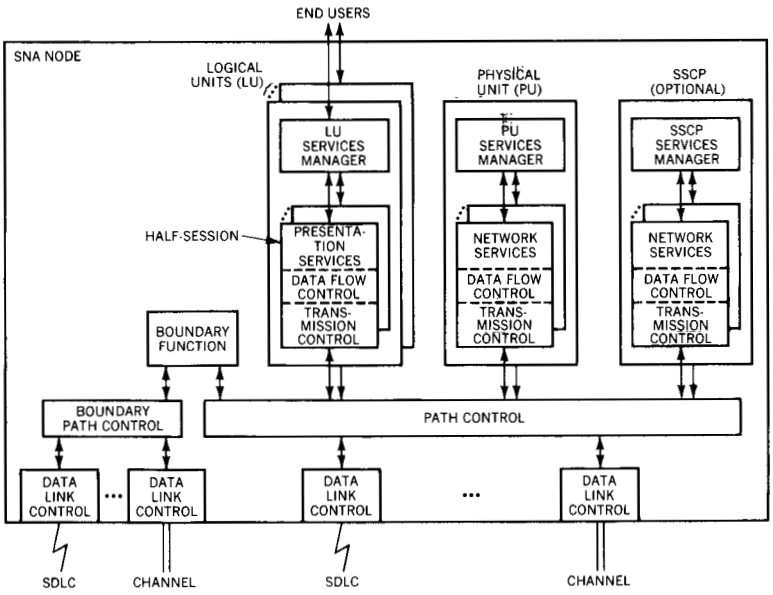
Systems Network Architecture (SNA) was announced by IBM in September 1974. Since then, the original set of functions, which supported single-system user data processing networks, has been enhanced to support multiple-system networking. Many new functions and improvements to the original set have been announced and installed on users' systems; they include SNA 2, SNA 3, SNA 4.1, and the latest, SNA 4.2.[1] This paper assumes some familiarity with SNA; SNA and its components are defined in References 1 through 4 and explained in References 2 through 16.

The first section of the paper is a brief review of SNA as it was designed for single-system tree-structured networks. This review includes the major elements of an SNA node and the operational characteristics of networks using SNA.

The second section reviews applications within multiple-system networks and discusses the requirements that these applications place upon the architecture. Included are discussions about the SNA enhancements that address these requirements.

Copyright 1979 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to *republish* other excerpts should be obtained from the Editor.

Figure 1 Structure of a node



The third section focuses on configuration services—the structuring and operation of the physical resources of the network. The fourth section focuses on session services—the initiation and termination of connections between network users. The fifth section discusses network operation, especially the identification of problems and their repair.

SNA for single systems

structure of a node

An SNA network is made up of nodes connected by data links. The behavior of the network as a whole is determined by requiring that each node in the network behave towards other nodes as if it were internally a subset of the model SNA node described in detail in Reference 2.

Figure 2 The simplest SNA network

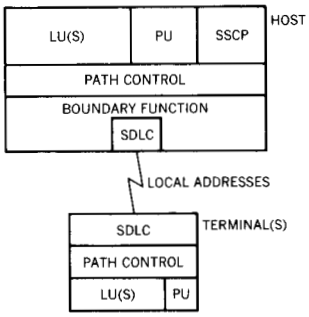
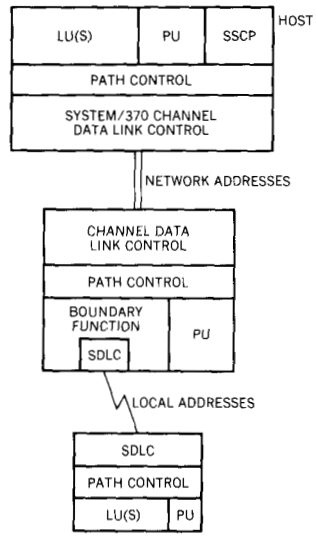


Figure 1 identifies the major elements of an SNA node. Each node contains a path control element for routing, as many data link control (DLC) elements to schedule transmission as there are link connections to adjacent nodes, and a Physical Unit (PU) to activate and control the links. There are a variable number of Logical Units (LUs); these act as ports into the network for end users.[17] Communications controller nodes (e.g., the network control program (NCP) known formally as ACF/NCP/VS, or Advanced Communication Function/Network Control Program/Virtual Storage) perform useful network routing and control functions without necessarily containing LUs, but most SNA nodes contain one or more LUs.

An SNA product may optionally implement a System Services Control Point (SSCP).[18] These control points provide two kinds of services to networks. First, they connect the network operator(s) to the PUs in the network. The connection between a control point and a PU is called an SSCP-to-PU session; this session allows activation, deactivation, and status monitoring of the resources of the network from network operator sites. Second, the control points coordinate the creation of sessions between LUs. Two of the services provided are: the resolution of LU names (used in logon requests from network users) to LU addresses (so that the network users are not sensitive to changes in network configuration) and allocation of access to LUs that are serially reusable. This LU allocation function is similar to device allocation in operating systems.[19] The resources (PUs, LUs, links, etc.) defined to a control point constitute its *domain*.

Figure 3 Another *single-system* network



Nodes may contain a boundary function. This is a point at which global network addresses are mapped into the local addresses used by adjacent terminals and cluster controllers. This mapping allows changes in network address assignments and system definitions without changes to the cluster controllers or terminals.

configurations

Figure 2 shows a single-system configuration: one host node with one or more terminals connected to it. The host node implements a control point; the other nodes typically do not. The IBM 3790 Communication System, System/34, System/38, DPCX and DPPX (the Distributed Processing Control Executive and the Distributed Processing Programming Executive on the IBM 8100 Information System), and VTAME (Virtual Telecommunications Access Method Extended on IBM 4300 processors with communication adapters) can be hosts in this configuration. Figure 3 illustrates a slightly more complicated configuration: a communications controller node (e.g., the network control program on the IBM 3705) has been inserted. By concentrating the traffic between the host and many SDLC (IBM's line control discipline called synchronous data link control[14]) lines, this node off-loads host processor cycles concerned with data link control management and improves the performance of the distributed system.

An enormous variety of configurations is possible with SNA;[20] some of the reasoning behind the architecture functions that make these configurations useful follows:

distributed processing

Networks exist to allow distribution of data processing applications. In the implementation and installation of single-system SNA networks, distribution has meant remote attachment of operator stations and remote execution of (usually simple) application programs.

Figure 4 Distributed processing

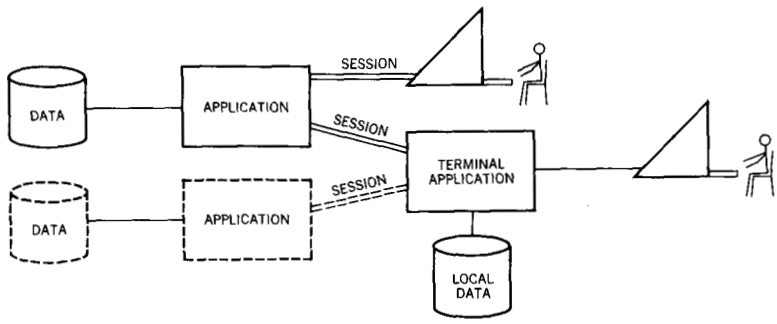


Figure 4 illustrates the distribution of function in a single-system network. Either the terminals are in direct session with an application program in the host, or a terminal application is placed near the terminal. The terminal application may access a limited amount of local data in order to handle some of the data processing locally. In the event of failure of the host or the path to the host, the terminal application may continue processing in a fallback mode.

Sessions between LUs are created when one LU (the primary LU, or PLU—often implemented in a host processor) sends a session activation request called a BIND to another LU (the secondary LU, or SLU). Parameters in the BIND request are used to tailor the properties of the resulting session to the capabilities of the LUs involved and the needs of the end users. A detailed discussion of these parameters is contained in References 2 and 3; here we mention the LU type parameter, which defines the behavior of each LU as seen by the other. This interface is product-independent so that a primary LU, for instance, can support LU Type 1 and thereby support an expanding list of products that implement secondary LU Type 1 interfaces. More information on LU types can be found in References 3 and 7 and in SNA product publications.

work station LUs

LU Type 1 appears to the primary LU to be a keyboard-printer console with alternate data destinations such as diskettes, punches, and extra printers. Implementation of alternate destinations is optional. LU Type 1 sessions have been used to support interactive keyboard-printer terminals (e.g., the IBM 3767), batch work station terminals (e.g., the IBM 3774, 3775, System/32, System/34), receive-only printers (e.g., the IBM 3287 and 3289), and print data sets (such as on the IBM 3790). High-performance batch work stations can use several LU Type 1 sessions in parallel; these "multiple LU" work stations are available in such products as the IBM 3790, the 3776 and 3777 terminals, and the DPPC and DPPX control programs. LU Type 4 is similar to LU Type 1 but, in order to meet different requirements, differs in some of the details of its protocols.[21]

display LUs

LU Type 2 presents the appearance of a keyboard-display operator's station. For example, the data stream of the IBM 3270 Information Display System differentiates it from the SNA character string data used with (most) LU Type 1 destinations; the scheduling of output is also different.[22] Secondary LU Type 2s specify an (optional) local copy capability in a way that makes the physical location of the "copy-to" device transparent to the primary LU. This independence of physical configuration contrasts with binary synchronous control (BSC) 3270 control units in which the physical clustering of display heads and printers is apparent to the subsystem that drives the display.

LU Type 3 is a receive-only printer that accepts the 3270 data stream. Its chief use is to print data that are normally displayed. A given printer can implement both LU Type 1 and Type 3 support, with the behavior of the session established at BIND time.[3,23] Like LU Type 2, LU Type 7 presents a keyboard-display appearance. In order to meet different requirements, it differs in the data streams that are accepted and in some details of its protocols.[21]

program LUs

A variety of programming and hardware items used in distributed processing (e.g., the IBM DPCX, DPPX, and Series/1) supply secondary LUs for application-program-to-application-program usage. The session protocols that are used vary depending upon those supported at the other LU. IMS (Information Management System), CICS (Customer Information Control System), and TCAM (Telecommunications Access Method), for instance, provide product-specific primary LUs for program-to-program usage.

Multiple-system applications and requirements

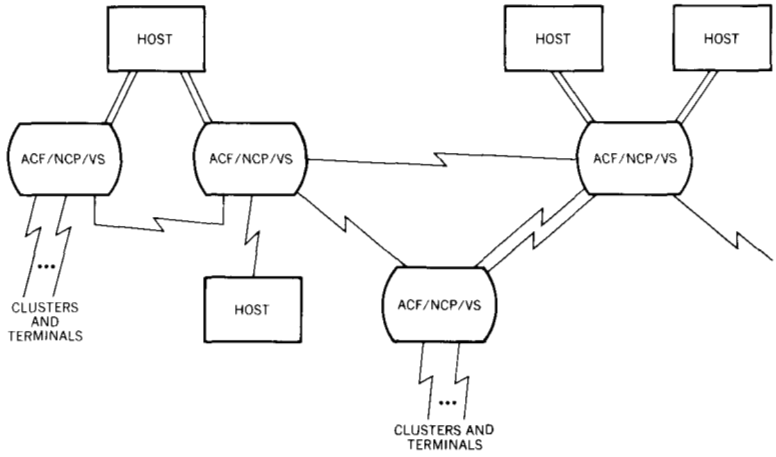
When multiple systems are connected into networks, such as that shown in Figure 5, processing can be distributed in many new ways. After sketching several such applications, we will review the network requirements created by the multiple-system environment.

batch applications

When the network contains two or more Job Entry Subsystem 2 (JES2) components, a network job entry application is available to the RJE (remote job entry) work stations. This application allows jobs to be submitted at a system with JES2 for execution at any JES2 system and routing of output to any JES2 system. (References 24 and 25 contain excellent discussions of network job entry and JES2.) The use of SNA offers the following advantages over the other JES2-to-JES2 connections:

- SDLC links operate in full-duplex mode. The BSC links supported by JES2 are only able to operate in half-duplex mode

Figure 5 Multiple-system networking



even though the basic transmission facilities provide simultaneous data transfer capabilities in both directions.

- In the non-SNA JES2 networks, it is necessary to either connect every JES2 node with every other JES2 node by a direct connection, or allow intervening JES2 nodes to provide store-and-forward routing of the transmitted jobs. When SNA is used, the JES2 LUs can be fully connected by sessions, thus eliminating the full job store-and-forward delay and processing overhead.
- When SNA is used, the job network can share communications controllers and links with other applications. This single network can be managed as an entity separate from any one of the applications that use its facilities.

Batch data transfers (other than network job entry I/O) may be desirable. These can either be applications under the JES2 network job entry facility or other subsystems (e.g., TCAM or CICS). Batch data transfers can often be done in a background mode, using residual transmission capacity left over from higher-priority applications. Or, if time zone differentials or scheduled unavailability of some network connectivity warrants, they can be done in several stages by store-and-forward-techniques.

Message-routing applications share many of the properties of batch data transfers, but the emphasis on ease of use for the terminal operators excludes JES2 from the choice of subsystems.

transaction routing

Now consider a transaction-processing application in which some transactions from each terminal are to execute at Host A, others at Host B. In the network job entry environment, JCL (job control language) statements specify the user's routing intent. In this case, a user-supplied or parameterized program analyzes the data

input with the transaction and determines whether to send it to A or B (or handle it locally). Some transactions may require a single input and a single output; others will require an extended conversation between the distributed application programs, running on many nodes, and the terminal operator. In order to simplify the design and coding of these more sophisticated applications, SNA has added LU Type 6—an LU-to-LU protocol especially suited to the transaction-processing environment. Also, in order to let extended conversations exist simultaneously, multiple parallel sessions between LUs have been added.

accessing remote data

It may be that the nature of the transaction to be handled does not require a sophisticated calculation, but rather, requires access to data that are not at the local node. Two solutions to this problem are possible: bring the data to the local node, or send the data base request to the remote node for execution by a transaction at that node. SNA defines, within LU Type 6, a general way to send the data base request to the remote node for execution. Sometimes referred to as "function shipping," this technique encapsulates the application's request, determines that it has to be executed remotely, and sends the request to the remote node for execution. The reply data are returned and presented to the application in the form expected by the application for replies to requests of that type. LU Type 6 defines the functions that can be "shipped" in terms of processes that are invoked within the LU that receives a request. These processes include ones for access to DL/I (Data Language/I), data bases, for access to message queues, for sending system messages, and for scheduling the execution of programs within the receiving LU. Implementation-defined processes are also possible, and CICS/VS[26] has implemented access to VSAM (Virtual Storage Access Method) and other datasets.

application to application

The same architecture definitions and product interfaces that permit transaction routing and access to remote data allow arbitrary distributed computations. This frees the application designer to place function where it is needed. This flexibility encourages application designers to layer and structure their applications and data bases so that new and changed function can be accommodated incrementally without extensive redesigns of existing applications.

One way to create modular designs is to use SNA sessions as interfaces. Perhaps the application is going to run on a single processor this year, but will it always? By dividing the application into two (or more) pieces that run attached to separate LUs, or are coupled by interfaces that support function shipping, the ability to run the application on two separate processors is created for the future. In the present, the interface between the two components will be firm and will be easier to monitor and control.

This kind of interface design is especially suited to interfaces between organizational entities (between departments, divisions, or companies).

requirements

By reviewing the network environment suggested in the previous discussions of applications, we see that multiple-system networks require additional function for:

- Connectivity. Certainly the architecture should support the connection of any two LUs whenever they have compatible capabilities, and SNA does this. Further, SNA products offer complementary ways to connect non-SNA products into an SNA network.[27-30]
- Usability. The network should be easy to install, repair, and change. Users should be insensitive to network changes.
- Resource Sharing. There should be shared use of resources, routing outside of larger hosts, and load balancing across the links of the network. The connections between nodes should be able to increase and decrease in effective speed by the addition and deletion of links as load requires.
- Availability. There should be parallel links between nodes, alternate routes between nodes, containment of the disruption resulting from failures, and nondisruptive resynchronization of network components during recovery from failures. The network should be robust, able to withstand failures of individual components, even able to withstand simultaneous failures.

The next three sections provide detail on the SNA features intended to satisfy these requirements.

Multiple system configuration services

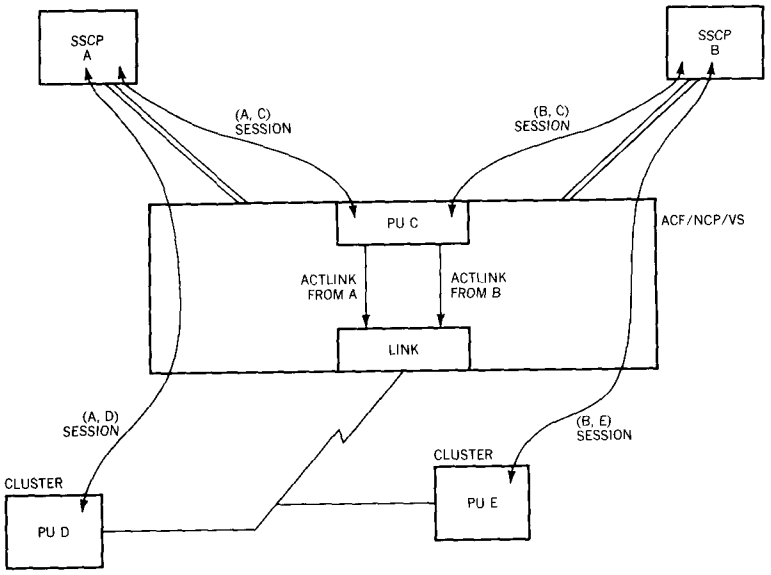
shared control

Multiple System Services Control Points may serially or simultaneously share control of the PUs, LUs, links, and link stations in the network. Associated with each PU, LU, link, and link station is a share limit that specifies the maximum number of control points that may concurrently share control.[18,31] Figure 6 illustrates one shared control configuration.

Shared control provides a mechanism that:[32]

- Gives the network designer freedom to establish which control points in the network may control PUs, LUs, and links throughout the network.
- Permits notification of a control point that another control point (or points) has been removed from the network because of failure or as a result of normal control point shut-down procedures.

Figure 6 Shared control of links



Shared control allows a network designer to define networks in which a control point in a single host is given responsibility for network configuration and maintenance control (the communication management configuration[7]) while other hosts contain application subsystems such as CICS, TSO (Time Sharing Option), or VSPC (Virtual Storage Personal Computing). In case of link or node failure, only the control point with network configuration responsibilities is notified of link or link station inoperative conditions. Session partners in the application hosts using the failing link or node for data transmission are notified of session damage (see the discussion on session outage notification later in this paper).

Alternatively, the network configuration may be defined to permit multiple control points to control specific parts or areas of a network. As an example, multiple control points may share control of a link that supports attachment of multiple cluster or terminal nodes, and each cluster or terminal may be controlled by any one of the control points sharing control of the link as illustrated in Figure 6.

Each control point need not be aware that others are sharing control of resources. The PU in each node is responsible for monitoring the share limit value of each sharable resource and prohibiting additional control points from obtaining control of these resources. A response denying control of a resource is sent to the control point requesting control if the share limit of the resource has already been reached.

The shared control function is also useful in control point failure or normal shut-down situations as a notification mechanism. If multiple control points are sharing control of a communications controller, and notification of the removal of a control point from the network because of its failure or network deactivation is sent to the operators at other control points in the shared environment, the operator at any control point that was previously defined as a backup for the one removed from the network may begin a resource-takeover procedure.

configuration insensitivity

Single-system SNA configurations were tree-structured, with a single control point in the root node. In SNA 3[1] this was improved; multiple trees could be interconnected near their roots. Specifically, local network control program (NCP) nodes could be interconnected, but remote NCP nodes could only be attached to a single local NCP node. This restriction on configuration had two origins in the implementations. One had to do with the packaging of the NCP load modules. When the remote NCP was first designed, storage was considerably more expensive than it is today, so the code was tailored to the specific configuration of a remote concentrator. No channel support was included. Only a single link could be made active. Similar restrictions existed in the local NCP packaging. Another restriction is discussed in the section of this paper that describes session outage notification.

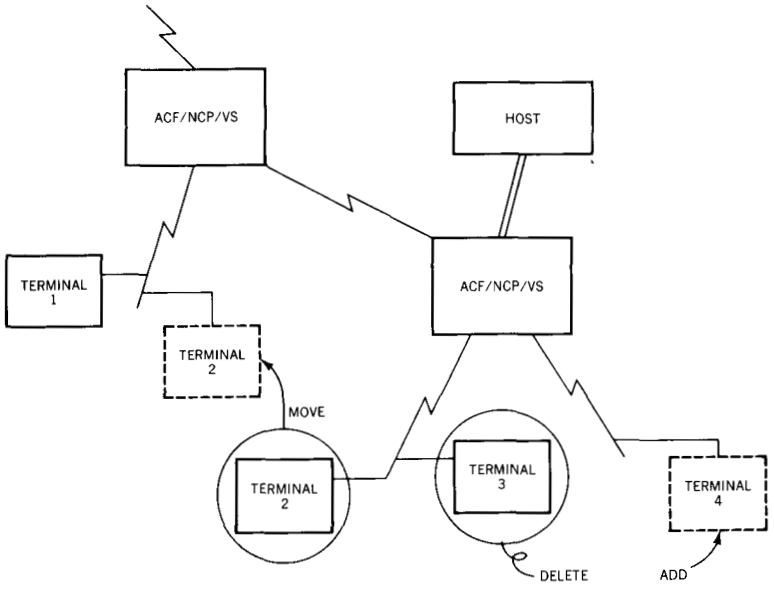
The SNA 4.2[1] release of NCP removed the SNA 3 configuration restrictions by integrating the packaging of NCP code; there are no longer local and remote versions. A single version of NCP adapts to the node on which it is to run. When NCP is disconnected from all external control points, it has a control point that monitors a set of links and connects to the rest of the network automatically. On links to other nodes, an exchange of identification data dynamically determines which node is to contain the primary SDLC station. This capability provides greater robustness for the network in the face of link disruptions and failures of nodes.

dynamic configuration

Dynamic configuration, as illustrated in Figure 7, allows clusters and terminals supported by a boundary function to be configured or reconfigured while the network is in operation. This function is important to networks that:

- Have a rapid increase in the total number of clusters and terminals added to the network or frequent movement of clusters and terminals.
- Wish to avoid the expense associated with frequent system generations.
- Support continuous operation with a low tolerance for the network disruption that may accompany static system generations.

Figure 7 Dynamic configuration



To dynamically configure the network, the network owner begins by defining the new configuration to one or more control points. This definition is done while the control points are on line.

A protocol between a control point and the PU (e.g., the NCP) supporting a boundary function is then used to configure the network. To add a cluster or terminal to the network the control point requests the PU to provide a network address for the cluster or terminal PU and LUs being added to the network. After the network addresses are assigned, the control point passes parameters associated with the cluster or terminal PU and LUs to the PU of the boundary function. An example of an LU-related parameter is the local address form of the LU network address. Now the control point can activate the dynamically configured PUs and LUs. LUs can also be dynamically added to statically defined clusters and terminals.

To delete a cluster or terminal from the network, the control point requests the PU of the boundary function to free cluster or terminal PU and LU network addresses.[33] To move a cluster or terminal, the delete procedure is invoked first, followed by the add procedure.

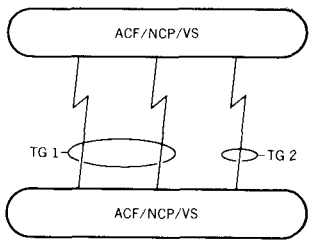
parallel links

Parallel links are defined as multiple links operating concurrently between the same two adjacent communications controller

nodes. Parallel links allow increased bandwidth whenever it is required and provide increased availability and reliability. Also, additional links may be added between adjacent nodes if tariff considerations make multiple slower-speed links less expensive than a single high-speed link or because the highest available speed of a single link does not provide sufficient bandwidth.

The control for parallel links was placed in the path control and physical unit services elements of SNA by introducing the concept of transmission groups. Transmission groups are defined by the subarea address pair of the adjacent nodes and a transmission group number. A transmission group represents one connection, or logical link, between adjacent subarea nodes. A set of parallel links may be divided into one or more transmission groups; this gives the network designer the ability to group links with similar characteristics (i.e., terrestrial, satellite, speed, quality, etc.) into a single transmission group. Path control routes (data from) sessions over transmission groups. See Figure 8.

Figure 8 Transmission groups (TGs)



Each link in a group uses its own SDLC protocol.[34] This permits simple addition and deletion of links, collection of error statistics for each physical link, and a scheduling algorithm that detects and compensates for degradation on one of the links. Traffic scheduled for transmission over a transmission group with multiple links is scheduled among the links[35] in order to best use the composite bandwidth of the links.

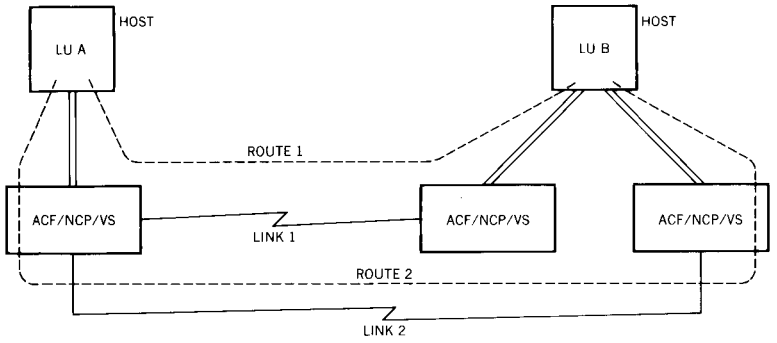
SNA session protocols require that all requests and responses arrive at the session end that is the destination in the same order they were transmitted from the session end that is the origin. But data blocks flowing over a multiple-link transmission group could arrive at the receiving end of the transmission group in an order different from the order transmitted by the sending end of the transmission group because of:

- Links operating at different bit rates with different propagation times
- Data blocks of various lengths transmitted across the transmission group
- Link errors that result in retransmissions

Any out-of-order blocks are reordered at the receiving end of each transmission group in the path. The transmission header contains a sequence number field that has a value set by the sending side of each transmission group and checked for correct order by the receiving side of each transmission group.[36,37]

A very desirable attribute of the multiple-link transmission group protocol is that a single link failure is not disruptive to sessions using the transmission group. Session traffic is automatically

Figure 9 Alternate routes



routed over the remaining links in the transmission group. However, notification of the inoperative link is sent to each control point that had activated the link.

The transmission group becomes operational when the first link contact procedure successfully completes, and the transmission group becomes inoperative when the last link discontact procedure completes or when the last link fails.

multiple active routes

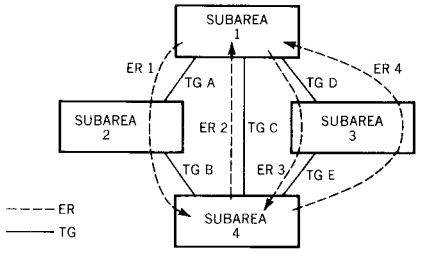
SNA allows more than one route between subarea nodes to increase the probability that a route will be available whenever an attempt is made to establish a session. If a route being used by a session becomes inoperative because of node or link failure, all session ends utilizing the failing route are notified; the session may be reestablished over another operational route. In Figure 9, if Route 1 supporting a session between LUs A and B becomes inoperative because of the failure of Link 1, the session may be reestablished over Route 2 that utilizes Link 2. Multiple routes can also be used to achieve load leveling, provide high security, or to keep batch traffic from interfering with interactive traffic.

The multiple-routing function uses two levels of control: explicit routes and virtual routes. To aid in network usability, the SNA user can request that sessions be given a (network level) class of service. Explicit routes and virtual routes, which are elements of the path control layer, will now be discussed, followed by a discussion of their interaction. Class of service is discussed in the next section on session services. More information can be found in References 2, 38, 39, and 40.

explicit routes

An explicit route is defined by: (1) the subarea address at one end of the explicit route, (2) the subarea address at the other end of the explicit route, and (3) an explicit-route number. The explicit-route number represents a specific sequence of transmission groups and subarea nodes connecting the subarea nodes at the end.

Figure 10 Explicit routes (ERs)



EXPLICIT-ROUTING TABLES

NODE 1

SA \ ER	1	2	3	4
2	TG A	–	–	–
3	–	–	TG D	–
4	TG A	–	TG D	–

NODE 4

SA \ ER	1	2	3	4
1	–	TG C	–	TG E
2	–	–	–	–
3	–	–	–	TG E

The following explicit routes are some of those that could be defined between Subareas 1 and 4 for the configuration in Figure 10 (TG means Transmission Group):

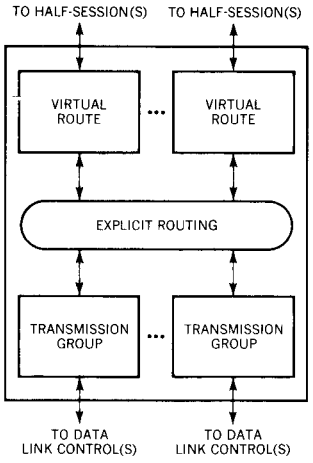
Explicit Route 1 = Subarea node 1, TG A, TG B, Subarea node 4

Explicit Route 2 = Subarea node 4, TG C, Subarea node 1

Explicit Route 3 = Subarea node 1, TG D, TG E, Subarea node 4

Explicit Route 4 = Subarea node 4, TG E, TG D, Subarea node 1

Figure 11 Inner structure of path control



The tables in Figure 10 are used by path control to direct data blocks to transmission groups based on the destination subarea and explicit-route numbers in the transmission header.[36] Notice that explicit routes are unidirectional, from an origin subarea to a destination subarea. Explicit routes are used in pairs that are physically reversible. In the example, Explicit Routes 3 and 4 are physically reversible. Explicit Routes 1 and 2 do not have reversed routes defined, although they could be added. The use of paired, reversible explicit routes simplifies failure notification in the network since it causes both directions of flow to fail simultaneously.

Virtual routes, explicit routes, and transmission groups (see Figure 11) form the elements of the SNA path control layer for subarea nodes. It is possible for multiple explicit routes to use a single transmission group. Explicit routes insulate the virtual-route layer from the physical configuration.

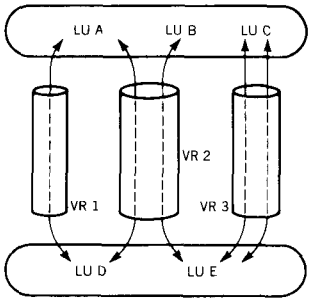
Explicit routes are used in SNA to allow alternate routing and load balancing.[41] In addition, they provide control over the physical

routing of session traffic: some traffic should not go over an insecure link (e.g., link-level cryptography is not installed), batch traffic should avoid the low-delay, but low-capacity, routes installed for interactive traffic, or interactive traffic should avoid long-delay routes.[42]

virtual routes

A virtual route identifies a full-duplex connection between two subarea nodes and only indirectly refers to physical connections. It is defined by: (1) a subarea address at one end of the virtual route, (2) a subarea address at the other end of the virtual route, (3) a virtual-route number, and (4) the transmission priority that is discussed in the last part of this section. A session is assigned to a specific virtual route at session establishment; multiple sessions may be assigned to the same virtual route. (See Figure 12.) When a session end is at a cluster or terminal node, a subarea node providing boundary function support contains the end of the virtual route used by the session.

Figure 12 Virtual routes (VRs)



A mapping of the virtual-route number to the explicit-route number is taken from a table when the virtual route is activated. Multiple virtual routes can use the same explicit route. During use of a virtual route, session traffic at the origin-subarea node is passed from virtual-route control to explicit-route control. Once the data block is passed to explicit-route control, it is passed from subarea node to subarea node using the explicit-route number and destination-subarea value contained in the transmission header. When the traffic reaches the destination-subarea node, it is passed from explicit-route control to virtual-route control. From virtual-route control, the traffic is passed to the session end that is the destination.

activating explicit and virtual routes

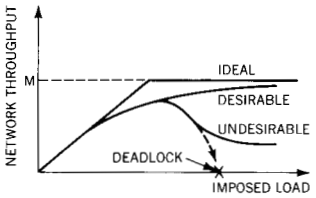
A distinction is made between operational and activated explicit routes. Explicit routes become operational because of node and link activations that make all the transmission groups in the explicit route operational. They become active when explicit-route activation messages flow across each transmission group within the specific explicit route.

When the transmission group becomes operational, requests flowing through the network declare appropriate explicit routes to be operational (see the discussion of session restart). Note that the propagation of these requests allows the users of the physical resources (links, nodes) of the network to queue on their availability (session activation as described below). Before an operational explicit route is first used for virtual-route traffic, an activation request is sent. It verifies that the explicit route is usable and complete (for instance, it detects looping routes and determines the route number of the physically reversible route) and measures the length of the explicit route (in units of traversed transmission groups or "hops").

Finally, the explicit-route activation protocols verify that no transmission group or node in the explicit route contains packets that have the subarea address pair of this explicit route; thus, use of the explicit route will begin at a known (reset) state.[43]

A virtual route is activated as a result of an attempt to activate a session. The selection of which virtual route to activate is based upon the requested class of service. If the first-choice virtual route is already active, the session activation request is transmitted on the desired virtual route. However, if the virtual route and associated explicit route are not in an active state, an attempt is made to activate the appropriate explicit route followed by activation of the virtual route.

Figure 13 Network throughput



M=MAXIMUM NETWORK THROUGHPUT UNDER IDEAL CONDITIONS

The explicit-route activation will fail if any subarea node cannot forward the explicit-route activation request because of network failures or incomplete configuration activations. When the first-choice virtual route and associated explicit route cannot be activated, an attempt is made to assign the session to the next virtual route in the list of virtual routes determined from the class-of-service name. (See the later discussion on class of service.) This algorithm repeats until the session is assigned to a virtual route or until it is determined that no virtual route in the class-of-service list can be activated. The unavailability of all virtual routes in the list results in a notification of failure to the LU initiating the session.

Activation of System Services Control Point sessions is slightly different. A class of service is used; explicit route and virtual route activation are also the same. The control point sessions, however, do not have to fail when no virtual route is available: they can be queued, pending the availability of a virtual route.

Class-of-service exits are provided to allow user-determined assignment of a session to a virtual route. These can be used to perform load balancing. With parallel sessions (described later), the exit can be used to establish simultaneous sessions over alternate routes, thus increasing system availability as perceived by end users.

flow control

Any network has a maximum throughput limit, which cannot be exceeded even if the network input traffic is unbounded. Due to cost considerations, commercial networks are normally designed so that peak network traffic loads occasionally exceed storage, cycle, and bandwidth capabilities of nodes and links within the network. Given this, SNA seeks to prevent significant network throughput degradation and to prevent network deadlock conditions as network load increases.[44,45] (See Figure 13.)

SNA provides global and local flow control mechanisms to maintain network throughput approaching the ideal as network loading increases. Global flow control utilizes virtual-route pacing to regulate the flow of traffic through the network. Traffic from many sessions on many virtual routes may be routed through the same physical nodes and links within the network. This means independent sessions have traffic within the network contending for the same storage, cycle, and bandwidth resources. When the virtual route is prevented from sending by its pacing algorithm, it queues session traffic until a virtual-route pacing response arrives indicating that adequate resources are available within the network to transport traffic across the virtual routes. This queuing for entry to the transit network ensures that it will not become overloaded.

Session pacing has characteristics similar to route pacing, but the rationale is different. The purpose of session-level pacing is to prevent one session end from sending data more quickly than the receiving session end can process the data.[46]

For virtual-route pacing, a pacing window of traffic (the number of additional packets that can be sent after a pacing response is received) is transmitted by one end of the virtual route after the other virtual-route end agrees to accept it. The initial window size is based on the explicit-route length measured during explicit-route activation. Performance analyses[47] have shown that dynamically adjusted window sizes can provide greater network throughput than statically defined window sizes. A static window size cannot take into consideration the amount of traffic and the changing resource availability within the network. Window sizes are adjusted for each virtual route by checking the amount of data enqueued at transmission group send queues and passing parameters in the transmission header indicating any required change to the current window size.[48]

Local flow control via session pacing is the mechanism used by a subarea node to regulate the entry of traffic (e.g., from an application or through a boundary function). The determination of when traffic can be accepted is based upon local buffer availability and the state of the virtual-route transmission queue for the session.

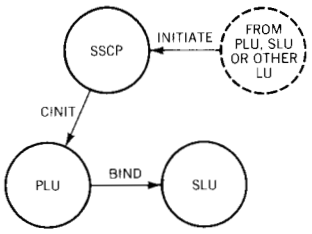
Two-stage inbound session pacing allows a boundary function to control the entry of traffic from cluster or terminal session ends. One stage of pacing is from the cluster or terminal session end to the boundary function, whereas the second stage of pacing is from the boundary function to the host. A pacing response to a cluster or terminal session end is withheld by the boundary function node, thus preventing specific cluster or terminal LU traffic from entering the boundary function,[49] if the boundary function

node is congested and if virtual-route traffic for a specific cluster or terminal LU cannot flow because the virtual-route pacing window has been exhausted.

transmission priority

Session traffic flows through the network at one of three transmission priority levels. Traffic at a higher priority is queued ahead of any lower-priority traffic at each transmission group send queue. This queuing is independent of the number of physical links associated with the transmission group. Within each priority level, traffic is queued FIFO (first-in, first-out) for delivery to an adjacent node. The FIFO queues are aged to ensure that lower-priority traffic is not completely stopped. Transmission priority is a property of virtual routes; sessions are assigned to virtual routes at session activation time and remain assigned for the duration of the session.

Figure 14 Session initiation within one domain



Low-priority traffic is displaced from the network when the amount of high-priority traffic increases because enqueuing the higher-priority traffic ahead of lower-priority traffic increases the round-trip delay for the latter. This slows the exchange of virtual-route pacing messages by delaying the pacing request and so reduces the amount of low-priority traffic admitted to the network. Because virtual-route pacing responses are so critical to network performance, they are transmitted at a fourth priority—ahead of all other virtual-route traffic, thus ensuring that heavy traffic in one direction will not interfere with the flow of virtual-route pacing responses in the other direction; if such interference had been allowed, it would have decreased network throughput under heavy loads. This is one way in which virtual-route protocols differ from link-level (e.g., SDLC) protocols.

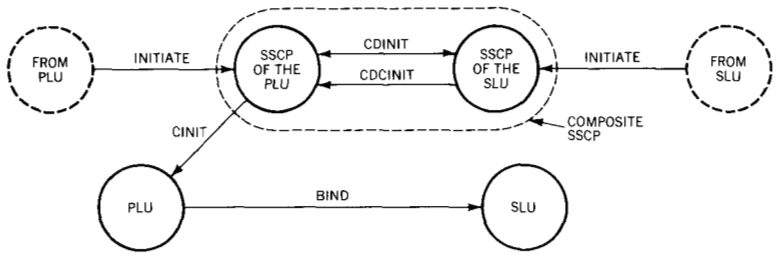
Transmission priority is useful for ensuring continued good response time to favored applications during periods of network overload. Also, since the network will displace low-priority traffic with higher-priority traffic, bulk data transfer applications can be run continuously instead of being scheduled for specific slack periods. These applications will utilize spare network capacity much as applications with a low dispatching priority utilize spare processor resources.[50]

Multiple-system session services

cross-domain sessions

The first implementations of SNA (e.g., VTAM Release 1.0, NCP Release 3.0) were limited to a single domain, while product design decisions further limited the network to a single System/370 node (each System/370 access method contained a System Services Control Point). The multiple-system networking capabilities of ACF/NCP/VS, ACF/VTAM, and ACF/TCAM made networks containing multiple domains possible.[7] One underlying extension to the ar-

Figure 15 Session initiation between two domains

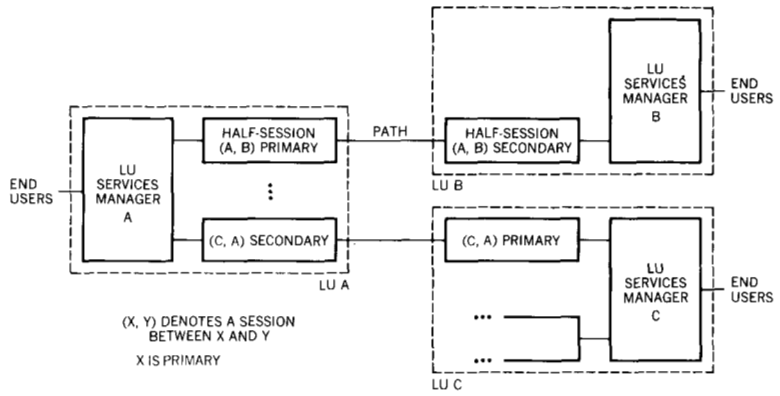


chitecture was the ability to initiate and terminate sessions between LUs in different domains. This ability means that a terminal LU, for instance, can be in session with an application LU in any host in the network.

When both LUs that are to be bound into a session are in the same domain, the sequence shown in Figure 14 establishes a session. The sequence begins when some LU sends an INITIATE request to the control point. This request can be entered in character-coded form by an operator at a terminal (e.g., a LOGON); in this case, either the LU or the control point will translate it into a formatted INITIATE request. The control point resolves LU names into network addresses, and when the session can be established, sends a CONTROL INITIATE request (CINIT) to the primary LU.[51] This request contains all the information the primary LU needs in order to send a BIND request to the secondary LU. Not shown in Figure 14 are additional requests that ensure synchronization of the three parties to the session under various errors and race conditions.

When the LUs are in two separate domains, two control points cooperate to provide session initiation services. As Figure 15 shows, this cooperation creates the appearance of a single control point as far as the LUs are concerned. Outside of the composite control point, the single domain sequence of INITIATE, CINIT, and BIND is unchanged. Inside the composite, the control point of the primary LU and the one for the secondary LU exchange requests that synchronize their behavior and accomplish the initiation. The control point of the initiating LU receives an INITIATE request and determines (from a table) that one of the requested LUs is in another domain. A CROSS DOMAIN INITIATE (CDINIT) is sent to the control point of this domain; the response to CDINIT carries the network address of the destination LU and the status of the session request (immediately available or queued). When the session is ready to start, the control point of the secondary LU sends a CROSS DOMAIN CONTROL INITIATE (CDCINIT) request; the control point of the primary LU sends a CINIT to the primary LU, and BIND flows to the secondary LU.

Figure 16 (LU, LU) sessions



When the destination LU is in the domain of the control point, has not yet been activated, and is reachable via a switched link, the control point will hold the response to CDINIT until the switched connection has been made, a network address assigned, and the LU activated. This process allows dial-out operations to proceed from the domain that results in the lowest connection costs without an impact on the application programs.

Each cooperating pair of control points in the network is connected by a single session. Session initiation requests for different LU-to-LU sessions are interleaved on this single session. The SSCP-to-SSCP sessions also carry session services requests for termination of a session and for handling errors and race conditions during initiation and termination procedures. More details can be found in Reference 2.

A number of objectives have been achieved in the design of session services for SNA:

- LUs are independent of the physical network configuration; they are also independent of domain boundaries.
- Domain boundaries, in turn, are established at the discretion of the user who can specify network configurations (e.g., the communications management configuration) that match the management needs of his organization.
- The session status of an LU can be displayed by its control point.
- LUs are synchronized, and access to them is allocated.

Further treatment of session services under network failures follows later in the discussion of session outage notification.

reduction of LU definitions

When a control point receives an INITIATE request, it needs a table containing definitions of cross domain LUs in order to be

able to send CDINIT to the proper control point. The one that receives CDINIT, however, does not need (other than for security access reasons) any definition of the LU that originated the INITIATE because all necessary information concerning it is carried in the CDINIT request.

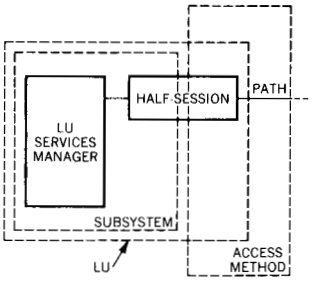
The SNA 4.2 products have used this fact to allow an optional reduction in the definition of LUs in the control point(s) that receive CDINIT. To see how this might work, assume that one control point manages all the terminals in a network[8] while several other hosts contain application LUs and subsystem LUs supporting application programs. For those terminals (e.g., displays) that are only used with operator-entered LOGONS, no LU definition is required in the control points in the application hosts.

Now let us consider another area of LU definition. It is necessary for the LUs in a session to agree to the parameters in the BIND that starts the session. In order to allow different applications to run well with the same terminal, several mode names may have to be defined to the control point of the terminal. The mode name in INITIATE will select a BIND image for the session. The definition of mode tables and corresponding definitions in subsystem LUs can be reduced for those products that support negotiable BIND. Negotiable BIND allows the secondary LU to return suggested BIND parameters to the primary LU if those contained in the BIND of the primary LU are not acceptable.

parallel sessions

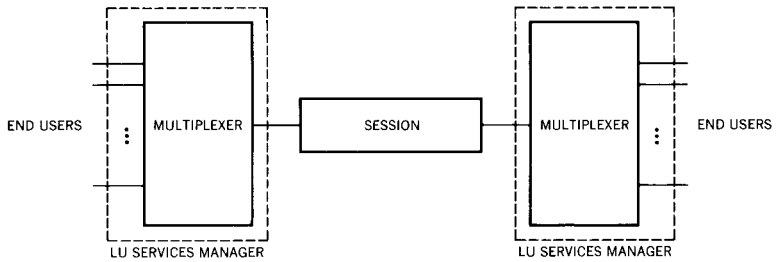
Figure 16 illustrates several possible sessions between LUs. We see that the session, consisting of a session end, or *half-session*, within each LU and a path between the LUs, is really between the two services managers (the services manager is the "center," or "core" of the LU; it connects end users to half-sessions, resolving contention as required).[52] One LU can contain both primary and secondary half-sessions. Initially SNA allowed at most one session between two LUs; this restriction existed because: (a) a session is identified uniquely by a pair of network addresses[53] (address of primary, address of secondary) and (b) each LU had a single network address.

Figure 17 Sessions and access methods



SNA has been implemented in a variety of ways in different products.[9,30,54,55] In some products, the LU services manager is in applications that use the access method (e.g., VTAM) and the half-sessions contained in the LU are split: parts are in the access method; parts are in the application. (See Figure 17.) Control blocks must exist to represent the LU to the access method and to represent the access method's portion of the half-session. Pointers are needed to connect the pieces of the half-session. In other products (e.g., TCAM), the half-session may be represented by a single control block.[56]

Figure 18 Multiple threads on one session



It may happen that a single session between two LUs is perfectly satisfactory; when one LU is a terminal (e.g., an IBM 3767 or a 3278), this is the case. But when the end users are programs served by a transaction-processing system, then many pairs of programs may need to be simultaneously connected. Figure 18 shows one way that this function can be provided: a single session can be used, with many threads multiplexed over it by the LU services manager. This approach is deficient in several ways:

- The session protocols (brackets, half-duplex, error recovery synchronization, etc.[2,3,10] perform needed functions. These protocols would have to be recreated on top of the base of the single session. This duplication would add extra overhead (for instance, a subsession request/response header would be needed), increase path lengths, and force the LUs to duplicate large amounts of lower-level SNA (e.g., access method) function.
- The individual threads would be invisible to the lower SNA layers. Thus, there would be no way to allocate different classes of service to different threads. Yet some threads will represent steps in high-priority transactions, others, just batch file transfers. Some will impose loads that are self-limited by operator input speeds; others will take all the capacity the network will provide. Some will need the security of encryption; others will not. Some will tolerate long delay routes; others will not.

If only one session is available between LUs, then failure of that one session will cause complete disruption. If, however, many sessions are available and if they can fail independently (see the earlier discussion of alternate routes), then system availability will be increased. If single sessions were used between LUs, the transmission subsystem would be required to deliver as much of the inherent (within the link and node topology) availability as possible to the single sessions. This requirement would have severely constrained the design of path control, thereby reducing the independence of design decisions within layers that should be independent.

Figure 19 Multiple LUs

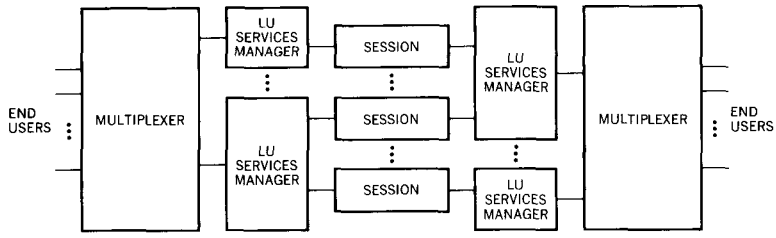


Figure 20 Parallel sessions

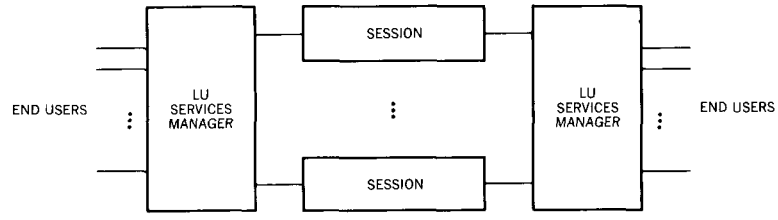


Figure 19 shows another way that multiple threads could be carried between subsystems: one or both session partners could implement multiple LUs in order to gain access to multiple network addresses. This approach, while implementing multiple sessions between subsystems, suffers from:

- Multiple LU names for one subsystem. The network operator would not have a convenient way to recognize or deal with the fact that the multiple LUs were really associated with a single subsystem. The operator of the subsystem would suffer similarly. Since the number of extra LUs could range up into the hundreds, what is a nuisance for several (e.g., RJE work station use of multiple LUs) could be an operational problem for others.[57]
- Every LU would have to incur the expense of the LU control blocks shared by the subsystems with the access methods. Since LU control blocks are typically many times larger than session control blocks, the expense would limit the number of sessions that could be used. But the system as a whole is clearly improved if sessions can be used as (nearly) free resources. Put another way: the correct design point for an SNA access method is many sessions per LU.

This leaves the true parallel session solution; it is diagrammed in Figure 20. The basic idea is to assign multiple network addresses

to an LU; new addresses are assigned as required to support additional sessions. While an optimal use of the address space could have been attempted, a constrained pattern of address assignments has been used to simplify the address assignment algorithms (especially in cross-domain session establishment). LUs are either capable of parallel sessions, or they are not. If they are not, they have a single network address, which is used for all sessions.[58]

LUs that support parallel sessions are given a single secondary address when activated. Whenever a primary address is needed, the control point asks the PU of the LU to assign one. Primary addresses can be used for multiple sessions with different LUs. When an address is no longer needed (all sessions using it have been terminated), the control point asks the PU of the LU to free it.[59]

Parallel sessions between LUs may have to be uniquely named for some uses. Uniqueness has been achieved by adding a primary half-session qualifier name and a secondary half-session qualifier name to BIND. Together they constitute the session qualifier name pair and identify a parallel session instance even if the network address changes. One use occurs when a session is being restarted after a failure: the correct checkpoint data need to be applied. The session qualifier pair is only shared between the LUs; it is not used by the control point(s) or network operator.

session outage notification

Both half-sessions must be notified of session damage (and then be reset) if node or link failures within the network disrupt traffic flow between the half-sessions. Without this notification, one or both half-sessions could enter a deadlock condition with no available session protocols to cause a reset of half-session information.[60] SNA notifies half-sessions of session damage by directing notification along the path of each session affected by the failure.

When a subarea node or transmission group fails, the nodes adjacent to the failing element detect a transmission group inoperative condition. This detection causes notification requests to be broadcast to each adjacent subarea node indicating which explicit routes are inoperative. A filter technique is used to prevent these requests from looping in networks with subarea nodes connected in loop configurations. The failure notification is propagated from node to node until it reaches the two ends of the explicit route, where a mapping is made from the explicit route to virtual routes, and the virtual routes are in turn declared inoperative. Session outage notification, which causes half-sessions to enter a reset state, is sent to each half-session assigned to the inoperative virtual route regardless of whether the session end is in a cluster, terminal, or subarea node. For LU-to-LU sessions, the notification is an UNBIND request.

When a boundary function link fails, the boundary function elements representing the LUs in the lost cluster or terminal are notified through boundary function path control and provide notification (by sending UNBIND) to the LUs on the other side of the subarea network. A link inoperative request flows to the control point that owns the link where it resets the sessions of the control point with the lost clusters and terminals.[61]

class of service

SNA users can request a class of service from the network for a session. As an example, some sessions may require a low response time, whereas others may require large bandwidth, more reliable connections, or more secure paths. Since the network users should be independent of the physical structure of the network, class of service is specified as a symbolic name in the INITIATE request. If the class-of-service name is omitted, it is derived from the mode name. If both are missing, a default is assigned. The class-of-service name resolves to a list of virtual routes; the session is assigned to the first virtual route in this list that can be activated.[62]

By proper definition of the virtual-route number to explicit-route number tables and careful definition of the class-of-service name to virtual-route list tables, each class-of-service name can mean the same thing for all users of the network.

Problem determination and recovery

communications network management

As the number of nodes in a network increases and the geographical area of the network increases, the problem determination aspect of networking becomes increasingly important. Communications network management applications[63,64] used with SNA allow a network manager or operator to monitor, identify, and isolate network problems from centralized points of control.[65] The centralized control philosophy is essential to allow quick and economical problem determination. Consider just a three-node network without central coordination: how does an operator attached to one node, working with a second, and routing data through the third, get an application working again when it fails? Only highly skilled operators (e.g., programmers) can be expected to do their own problem determination, and even they will take a long time to get it done. The centralized problem determination philosophy is a natural result of the "help desk" concept: the terminal operator should be given a single number to call when his application does not work. Roughly 85 percent of all calls to the help desk will be application-related;[66] some of the rest will lead to problem determination and recovery action. One goal of the problem determination facilities is to identify problems before they cause user complaints.

Error-oriented data may be sent to the communications network management application by using a solicited or unsolicited message. The System Services Control Point delivers unsolicited SNA data to the problem management application. However, the volume of error data flowing in the network can be more easily controlled by using a solicitation protocol as the primary mechanism for gathering data.

Certain errors are classified as permanent because once they occur the failing element is no longer operational. Other errors are classified as temporary, such as some communication line errors; for these the link is not declared inoperative until a temporary error count threshold has been reached. The temporary error data are especially useful in the detection of deteriorating lines. When a suspicious link has been identified, an intensive mode of error recording can be used to closely monitor the link. Further, nondisruptive link tests can be invoked. Of course, one of the most useful SNA features is the pause and retry logic that allows transient link failures to be waited out without session damage.[13,67]

Usually it is not sufficient to isolate problems to a specific physical node. Once the physical node has been identified, further analysis is required, perhaps at a remote site. Problem determination provides help in isolating a network problem to both the SNA network element and the physical element, such as a modem, line scanner, or control unit.

Other tools are available to aid in network problem determination. Links can be traced individually, transmission groups can be traced, sessions can be traced (within access methods). Explicit and virtual routes can be tested. Dynamic dumps of the network control program can be taken. Shared control of links can be used to ensure reporting of failures to both a primary and a backup network management center.

repair and reconfiguration

After a problem has been identified, repair of the failing component can proceed. When availability objectives require it, service can be maintained through reconfiguration of the network. SNA systems provide several aids to reconfiguration.

- Switched network backup allows failing leased lines to clusters or terminals to be replaced by switched network lines. This reconfiguration, which can be invoked by a programmed network operator, is not visible to the LUs of the network.[68]
- Re-IPL (initial program load) of failing network control program nodes can also be invoked by the network operator.
- Backup control points can take control of all or part of the domain of one that is failing. Control point restart and takeover can be nondisruptive; more discussion follows below. A

failing control point can reacquire its domain, but this may involve restart of LU-to-LU sessions.

- Applications can be moved transparently to different nodes, due to the configuration insensitivity provided to LUs by session services.
- Distributed applications can continue in local fall-back mode.

Repair and reconfiguration of the physical network are followed by restart of the damaged sessions which is now discussed in more detail.

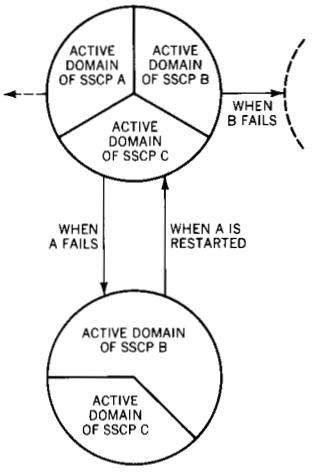
control point restart and takeover

Many LU-to-LU sessions may remain active when the System Services Control Point controlling an LU fails or when elements in the path connecting the control point and LU fail. As an example, a cross-domain session involving a cluster controller may remain active even if the control point controlling the cluster fails. The cluster would not be under the control of a control point because the SSCP-to-cluster PU and SSCP-to-cluster LU sessions would be reset by session outage notification.

Restart occurs when a control point that previously controlled resources, but lost this control because of network failures, reassumes control of those same resources. The restart protocols may be initiated immediately upon availability of a control point and route.

Takeover occurs when a control point obtains control of resources that were lost by another because of network failures. Figure 21 illustrates this process. Whenever the communications controller node PU detects damage of a session with an owning control point, notification that a control point has been lost is sent to all others sharing control of the communications controller node PU. The notification of the lost control point is a signal for the network operator at any control point with resource takeover responsibilities to initiate the proper resource takeover protocols. All control point restart and takeover protocols are identical to normal network bring-up protocols.[69] When restart or takeover for a control point occurs, it is desirable to take control of resources such as cluster PUs and LUs without resetting any active cluster LU to host node LU sessions. This is done by a nondisruptive form of the SSCP-to-PU and SSCP-to-LU session activation requests that does not reset LU-to-LU sessions (for those clusters that support this function).

Figure 21 SSCP takeover



session restart

Once physical repair has restored connectivity, sessions can be restarted. The first sessions restarted are between control points and their PUs, their LUs, and other control points. These sessions are queued on the availability of the relevant virtual routes. Sessions between LUs are restarted in a variety of ways, in some

cases automatically (e.g., TCAM), in other cases through an INITIATE request from the primary LU or secondary LU.

The restarted sessions invoke the usual class-of-service algorithm; the result is that an alternate virtual route will be found for the session to use if one has been specified and it can be activated. The installation can choose to provide alternate routes only to those applications where this is cost-justified.

When the sessions have been restarted, the session partners have to be resynchronized. The SSCP-to-PU, SSCP-to-LU, and SSCP-to-SSCP sessions are resynchronized using specific detailed features of the normal network services protocols used on these sessions. LU-to-LU sessions are resynchronized in three different ways:

1. Operator action may be required. This is typically the case with sessions that involve terminal operators. The IMS or CICS operator, for instance, may enter an inquiry to determine the status of the transaction entered prior to the failure. The output message from a transaction may be saved ("protected") by the primary LU so that when the terminal operator receives it, the completion status of the last transaction will be apparent. Other examples include backing up N pages in an interrupted remote job entry print file.
2. Product-specific protocols may be invoked. TCAM, for instance, can retransmit a TCAM message that has not been acknowledged. The receiver discards it if it has already been received successfully. Network job entry for JES2 also uses a product-specific protocol during resynchronization.
3. The sync point protocol of SNA may be invoked. LU Type 6 sessions and certain other sessions supported by IMS and CICS take a distributed sync point (or check point) during session processing. Whenever a failure occurs, the session partners roll back to the most recently completed sync point. At this point, the operation may continue by automatically restarting the partially completed work (e.g., IMS), or restart processing may have to be programmed by the customer (e.g., CICS).

coexistence of releases

Nodes that implemenent SNA 3, 4.1, and 4.2 may coexist within the same network. Transmission headers are converted between SNA 3 or 4.1 formats and the SNA 4.2 format as required. Sessions that involve SNA 3 or 4.1 nodes (i.e., in transit, or as control points, or that contain one or both LUs) will be unable to take advantage of most of the additional SNA 4.2 function.

Summary

The central theme of this paper is that SNA provides network services that simplify the design, implementation, and operation

Table 1 Product matrix

Function	*Product*					
	NCP	*VTAM*	*VTAME*	*TCAM*	*TPNS*	*ACP**
System/370 channel	3	1	ACF 1	10		8
SDLC						
Leased	3		ACF 1		4	
Dial	4		ACF 1			
FID 1	3	1	ACF 1	10	4	8
2	3	2	ACF 1	ACF 2.1	1	
3	4		ACF 1		1	
4	ACF 3	ACF 3		ACF 2.3		
Cross-domain		ACF 1	ACF 1	ACF 1.1	4	ACF
Parallel sessions		ACF 2	ACF 1			ACF
Path SON	ACF 3	ACF 3		ACF 2.3		ACF
Shared control	ACF 1	2	ACF 1	ACF 1.1		ACF
Dynamic configuration	ACF 2	ACF 2		ACF 2.1		
Parallel links	ACF 3					
Configuration insensitivity	ACF 3	ACF 3		ACF 2.3		
Transmission priority	ACF 3	ACF 3		ACF 2.3		
SSCP restart/takeover (nondisruptive)	ACF 3	ACF 3		ACF 2.3		
Multiple routes	ACF 3	ACF 3		ACF 2.3		

*Airline Control Program

Notes:

1. References are those of the first release of the component to support a particular function. ACFn means, for example, ACF/NCP Release n, or ACF/VTAME Release n, depending on the column in which the reference appears. ACF n.y under TCAM means ACF/TCAM Version n, Release y. ACF under ACP means ACP/TPF-ACF. A function is only listed if directly supported by the node in which the product runs. For instance, ACF/TCAM 1.1 supports SDLC dial lines, but the lines are attached only to NCP.
2. Only products or components that support cross-domain session services are shown. Many products support session services in single-domain and pass-through configurations.
3. Many features and functions implemented by the products are not listed in this summary chart. For instance, configuration restart is a service provided to the network operator by some products; it automatically reestablishes the same activation state in the domain of the control point as existed prior to a control point failure. Another example is the use of explicit routes without defining them to the product (available, e.g., in ACF/TCAM Version 2, Release 2). Or consider the elimination of the requirement that the names of origin LUs be defined to the control points of destination LUs (e.g., in ACF/VTAM Release 3 and ACF/TCAM Version 2, Release 3). Still another example is the support of certain non-SNA terminals through SNA sessions supported by LUs in NCP.[27]

of distributed processing applications. It does this through support of:

- Twenty-four-hour operation. The reduction of System Services Control Point cross-domain LU definition, multiple active routes, dynamic configuration, shared control, and the self-adjusting algorithms that perform flow control are all directed to reducing the number of times when a portion of the network needs to be taken off line for modification.

- Any configuration. Cross-domain session services, shared control, dynamic configuration, parallel links, configuration insensitivity, multiple active routes, and control point takeover all enlarge the user's choice of network configurations. The extra degrees of freedom that this provides can be used to increase operating convenience, increase availability, and reduce costs.
- Usability and availability. Communications network management, session outage notification, control point restart and takeover, session restart, multiple active routes, class of service, and flow control all improve the usability and availability of the network.

The growth of the network architecture to provide many functions was discussed. Since products implement only those functions needed by their users, not all products have implemented all functions. A partial summary of product support of SNA is contained in Table 1.

ACKNOWLEDGMENTS

Many people have contributed to the definition of SNA; the following have been chiefly responsible for that portion of SNA discussed in this paper: Fred George, Jackie Jackson, and Larry Loucks from TCAM; Jeff Knauth, Barbara Heldke, Frank Brice, Jay Benjamin, Ralph Naylor, Bob Weingarten, Art Jaeger, Jim Gilman, John Oseas, Roland Beauchaine, and Larry Area from VTAM and Communications Systems Programs; Glenn Huff, Stan Dilley, Miriam Green, John Eisenbies, Fred McGriff, and Carol Scopinich from NCP; Jim Gray, Philippe DeBacker, Marc Levillon, Nishan Bouroudjian, Tony McNeill, Mike Doss, Gene Thomas, George Plotsky, Vijay Ahuja, Bill Schaal, Bill Abrahams, and George Deaton from Communication Systems Architecture; and Martin Reiser and Kiyoshi Maruyama of the Research Division. We also thank Gary Schultz for his critical review.

CITED REFERENCES AND NOTES

1. SNA 2 is a short way to refer to some or all of the SNA functions contained in NCP/VS Releases 4 and 5, VTAM Release 2, and TCAM Release 10. SNA 3 is a short way to refer to some or all of the SNA functions contained in ACF/NCP/VS Release 1, ACF/VTAM Release 1, and ACF/TCAM Version 1. SNA 4.1 is a short way to refer to some or all of the SNA functions contained in ACF/NCP/VS Release 2, ACF/VTAM Release 2, and ACF/TCAM Version 2, Release 1. SNA 4.2 is a short way to refer to some or all of the SNA functions contained in ACF/NCP/VS Release 3, ACF/VTAM Release 3, and ACF/TCAM Version 2, Release 3.
2. *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic*, SC30-3112; available through the local IBM Branch Office. This is the definitive reference for details of SNA.
3. *Systems Network Architecture: Logical Unit Types*, GC20-1868; available through the local IBM Branch Office. This contains definitions of LU types.
4. *IBM Synchronous Data Link Control General Information*, GA27-3093; available through the local IBM Branch Office. An introduction to SDLC.

5. *Systems Network Architecture General Information*, GA27-3102; available through the local IBM Branch Office. An overall introduction to SNA as it was defined at its original announcement.
6. *Systems Network Architecture: Types of Logical Unit Sessions*, GC20-1869; available through the local IBM Branch Office. This is an introduction to LU types.
7. *Introduction to Advanced Communication Function*, GC30-3033; available through the local IBM Branch Office. This is a general introduction to SNA multiple-system data communication networking.
8. T. F. Piatkowski, D. C. Hull, and R. J. Sundstrom, "Inside IBM's Systems Network Architecture," *Data Communications*, 33-48 (February 1977).
9. H. R. Albrecht and K. D. Ryder, "The Virtual Telecommunications Access Method: A Systems Network Architecture perspective," *IBM Systems Journal* **15,** 53-80, No. 1 (1976). SNA as implemented in VTAM release 2.0 (SNA 2).
10. D. J. Eade, P. Homan, and J. H. Jones, "CICS/VS and its role in Systems Network Architecture," *IBM Systems Journal* **16,** No. 3, 258-286 (1977). Advantages of SNA to CICS/VS.
11. E. H. Sussenguth, "Systems Network Architecture: A Perspective," *ICCC 1978 Conference Proceedings*, Kyoto, Japan (1978).
12. J. P. Gray and C. R. Blair, "IBM's Systems Network Architecture," *Datamation* **21,** No. 4, 51-56 (April 1975). An overview of SNA that stresses user requirements and benefits.
13. J. P. Gray, "Network services in Systems Network Architecture," *IEEE Transactions on Communications* **COM-25,** No. 1, 104-116 (January 1977). A description of network services in SNA 2.
14. R. A. Donnan and J. R. Kersey, "Synchronous data link control: A perspective," *IBM Systems Journal* **13,** No. 2, 140-162 (1974). The requirements for, and benefits of, SDLC.
15. J. H. McFadyen, "Systems Network Architecture: An overview," *IBM Systems Journal* **15,** No. 1, 2-23 (1976). An overview of SNA 2.
16. W. S. Hobgood, "The role of the Network Control Program in Systems Network Architecture," *IBM Systems Journal* **15,** No. 1, 39-52 (1976). SNA as implemented in NCP/VS Release 4.0 (SNA 2).
17. The end users are outside the SNA node. Their relationship to the SNA node is defined by the product that implements the SNA node. The SNA node is an abstraction, with no necessary relationship to a physical node. For instance, ACF/VTAM and ACF/TCAM running in different regions of MVS represent two separate SNA nodes.
18. A product that does not implement a System Services Control Point (SSCP) implements a PU Control Point (PUCP). The SSCP is inside the SNA node; the PUCP is outside of it. The PUCP contains a small subset of SSCP function; it is used to activate the link(s) that connect the product to an SSCP(s). Small products use highly optimized PU and PUCP designs totaling a few hundred instructions. ACF/NCP/VS Release 3, however, contains a complete PU, a PUCP to activate its channels, and an SSCP to activate its SDLC trunks in the absence of other SSCPs. Products that contain SSCPs do not have to have a PUCP. A PUCP is not included in the share limit for a PU, link, or link station.
19. From one view, SNA provides an operating system for loosely coupled multiprocessors (this view was suggested by G. D. Schultz in 1973). Within this analogy, a session initiation is similar to "job" initiation, each half-session is similar to a subtask.
20. One interesting configuration (implemented by the DPCX and DPPX control programs on the IBM 8100 system) connects two independent SNA networks with an application program between two LUs, one LU in each network. The application can transparently pass data between the LUs. The pass-through application may become visible during session initiation, sometimes requiring a double "logon."

21. A discussion of the reasons for differences between superficially similar LU types (e.g., Types 1 and 4) is beyond the scope of this paper. LU Types 4 and 7 have been implemented by IBM products such as the System/34, System/38, and the 5250 family of terminals.
22. Whereas LU Type 1 assumes that session pacing provides all required source-to-destination synchronization in both directions, LU Type 2 specifies that primary LU to secondary LU traffic must be synchronized at the end-user level. Transmission of successive screens of output, then, must be interlocked with an operator keystroke (e.g., a program function key). The secondary-to-primary LU traffic is synchronized by pacing and by the restriction that each input chain end with a change of direction. This restriction of a general capability reflects a general layering principle; it allows, for instance, a common data flow control module to work across all SNA half-sessions. For a product application of this, see CICS.[26]
23. *IBM 3270 Information Display System Component Description*, GA27-2749; available through the local IBM Branch Office. Chapters 7 and 8 and Appendix F describe the 3270 product support of LU Types 1, 2, and 3.
24. R. P. Crabtree, "Job networking," *IBM Systems Journal* **17,** No. 3, 206–220 (1978). An overview and history of network job entry facilities.
25. R. O. Simpson and G. H. Phillips, "Network job entry facility for JES2," *IBM Systems Journal* **17,** No. 3, 221–240 (1978). Design objectives, implementation, and extensions of JES2 networking job entry.
26. CICS/VS *Version 1, Release 4 System Programmer's Guide (OS/VS)*, SC33-0071; available through the local IBM Branch Office. Includes material on CICS support of LU.T6.
27. *Network Terminal Option (NTO) General Information Manual*, GC38-0297; available through the local IBM Branch Office.
28. *Series/1 SNA Macro Programmer's Guide*, SC34-0228; available through the local IBM branch office.
29. *Distributed Processing Programming Executive (DPPX) General Information Manual*, GC27-0400; available through the local IBM Branch Office.
30. *TCAM General Information Manual*, GC30-3057; available through the local IBM Branch Office.
31. LUs and PUs attached through a boundary function have a share limit of one. This means that sharing must be serial. ACF/NCP Release 3 allows share limits greater than one on trunks. This allows an SSCP to share ownership of the physical resources along the entire path to a shared PU, thus preventing accidental deactivation of these resources.
32. In the SNA 3 implementations there was another reason to used shared control: The channel data link control protocol was not fully separated from activation of the SSCP-to-PU session. Each access method that wanted to activate a channel data link control to ACF/NCP/VS had to issue an ACTIVATE PU request over that channel.
33. The request to free PU and LU network addresses can be sent from an SSCP for statically defined clusters and terminals but must be sent from the same SSCP that requested network addresses to be assigned for dynamically defined clusters and terminals; this prevents accidental deletion of resources. A request to free LU network addresses will be honored if all sessions with the LU to be removed from the network are inactive. The request to free PU network addresses will be honored if all sessions with the node to be removed from the network are inactive and if all link procedures for the link connecting the boundary function to the cluster or terminal are inactive.
34. As compared to alternate methods that place parallel physical connections under one link protocol, SNA parallel links provide on-line (dynamic) addition/deletion of individual links to the parallel link group. Problem determination and repair is also better since links are individually controlled by the SSCPs and communication network management applications, leading to higher availability.
35. The transmission group frame-scheduling and error-recovery protocols assume that all the links in a group have approximately the same speed and

transmission delay. Transmission groups that are inhomogeneous will create more out-of-order data blocks. This will increase re-FIFO storage requirements and will raise the average delay across the transmission group toward the delay of the slowest link in the group.

36. The transmission group re-FIFO, explicit-route, virtual-route, and transmission priority functions use parameter fields in format 4 transmission headers. This format was introduced with SNA 4.2.
37. Support by SNA 4.2 nodes of SNA 3 and 4.1 nodes is required to permit smooth release-to-release transitions. Since SNA 3 and 4.1 nodes require that data blocks be delivered in order, yet do not contain re-FIFO logic, TG re-FIFO is required to simplify the migration support.
38. V. Ahuja, "Routing and flow control in Systems Network Architecture," *IBM Systems Journal* **18,** No. 2, 298–314 (1979), this issue.
39. R. R. Jueneman and G. S. Kerr, "Explicit path routing for switching network," *IBM Technical Disclosure Bulletin* **18,** No. 9, 3054–3062 (February 1976). The basic concept of explicit routing.
40. R. R. Jueneman and G. S. Kerr, "Explicit path routing in communication networks," *Proceedings of the Third International Conference on Computer Communications*, Toronto (August 1976). More on explicit routing.
41. W. L. Price, "Data network simulation experiments at the National Physical Laboratory 1968–1976," *Computer Networks* **1,** No. 4, 199–210 (1977). This study shows that dynamic routing schemes can have difficulty in load balancing.
42. F. H. Moss and P. M. Merlin, *A Routing Scheme for Session-Oriented Store and Forward Computer Networks*, Research Report RC 7427, IBM Corporation, Research Division, Yorktown Heights, NY (November 10, 1978). Reports work that might allow automatic generation of explicit routes when the topology of the network changes. When a routing algorithm only has to deal with single blocks, such as jobs in JES2 network job entry,[25] dynamic routing table updates are simpler to accomplish than in session-oriented networks such as SNA.
43. Explicit-route operational and activate protocols were introduced in SNA 4.2. In SNA 3 and SNA 4.1 there is only a single route between subarea pairs; after a failure it is not explicitly reset but goes to a reset state as data blocks flow out of it. If the route becomes operational very soon after a failure (e.g., a link is reactivated very quickly), the route should not be used until it is reset since hang conditions (e.g., lost responses) and lost data can occur as a result of race conditions between the blocks received before and after failure. This situation, not serious in the simple SNA 3 and SNA 4.1 networks, would have created difficult-to-manage operational procedures in SNA 4.2 networks. This follows since a direct correlation between a link failure and explicit-route failures is not visible to the network operator. Indeed, this correlation is known and used in only two places: the collection of active path control routing tables, and the central network routing design center. Explicit-route activation protocols eliminate these race conditions: an activated explicit route contains no data blocks from previous activations. The explicit-route operational protocols summarize that portion of the status of the entire network that is relevant to a given node and delivers it to that node.
44. This section discusses management of the imposed load. Additional control can be expressed by withholding load entirely, e.g., by scheduling batch transfers for third shift.
45. G. A. Deaton and D. J. Franse, "A computer network flow control study," *ICCC 1978 Conference Proceedings*, Kyoto, Japan (1978).
46. Until SNA 4.2, session pacing also served to provide global flow control. Virtual-route global flow control, introduced in SNA 4.2, allows significantly higher levels of link utilization to be achieved (in many configurations) before the knee in the curve of response time versus load is reached. The decoupling of LU buffer management (i.e., session pacing) from transit network scheduling (i.e., virtual-route pacing) also makes the network easier to configure (especially since virtual-route pacing is self-configuring). Notice that session

pacing limits the flow of normal flow requests; it does not govern the flow of responses or expedited flow requests. Session pacing, then, can be withheld for long periods without damage to the network—expedited commands such as UNBIND or SIGNAL can get through. Virtual-route pacing governs the flow of all session traffic; as a result, virtual routes cannot stay blocked for extended periods.

47. G. A. Deaton, "Flow control in packet-switched networks with explicit path routing," *Proceedings of the Flow Control in Computer Networks Conference*, Paris, France (February 12-14, 1979). This paper discusses flow control schemes similar to those used in SNA 4.2.
48. While global flow control greatly reduces the frequency of occurrence, the buffer resources in a node may still become depleted. Before the buffers are entirely used, the links will use the SDLC Receive Not Ready supervisory frame to reduce the rate at which traffic is admitted to the node until more buffers are again available.
49. If a session was bound with inbound pacing off, then the entire cluster controller will be polled with Receive Not Ready when the boundary is congested or the virtual route is blocked.
50. Notice the interaction of transmission priority with flow control: without virtual-route pacing, a surge of high-priority traffic would not displace low-priority traffic in the transit network; it would merely increase the lengths of the lower-priority queues. This could lead to buffer depletion; slow-down algorithms would be invoked (to avoid buffer depletion deadlock); and total network throughput would decrease.
51. The SSCPs may change the network configuration by establishing a switched connection to an LU. Also, the INITIATE request can be queued on the availability of an LU. The SSCP of the secondary LU also provides a table (indexed by the mode name field in INITIATE) that contains a suggested BIND for use on the session.[13]
52. The LU services manager has two pieces—one for (LU, LU) sessions, the other to help the SSCP in session initiation and termination. Most or all of the latter component is packaged inside the access methods.
53. Actually, a session definition requires a triple (address of primary, address of secondary, virtual-route identifier) to resolve certain races that can occur during session activation on one virtual route while session outage notification or deactivation is occurring on another virtual route. Once activated, the session is uniquely defined by the address pair.
54. *VTAM General Information Manual*, GC27-0462; available through the local IBM Branch Office.
55. *ACP/Transaction Processing Facility Concepts and Architecture Manual*, GH20-2157; available through the local IBM Branch Office.
56. In VTAM the half-session control block is the FMCB. When running with CICS, the CID and USERFLD pointers link it to the TCTTE of CICS. The VTAM ACB represents the LU. In TCAM, the half-session control block is the TTE.
57. TSO on VTAM uses multiple LUs to gain the benefits of the isolation that one LU per MVS region provides. The network operator can easily reference each session by the name of the terminal LU since it has a session limit of one.
58. Some limited parallel-session capability (e.g., for secondary half-sessions only) within path control is technically possible for such LUs, but since they would not (generally) support the INITIATE and BIND format extensions that are needed by parallel-session LUs, some products may choose not to implement this support.
59. The same requests, REQUEST NETWORK ADDRESS ASSIGNMENT and FREE NETWORK ADDRESSES, are used in dynamic configuration, DIAL, and parallel-session protocols.
60. Session failure timers could be used, but setting their values intelligently would require knowledge of the network configuration and load. This would

constitute an undesirable violation of the principle that LU protocols should be independent of the network physical configuration. Also, timers are expensive to use in quantities.

61. In SNA 3, session outage notification was performed with the help of the SSCPs. This meant that an LU-to-LU session outage signal between two SSCPs could fail to arrive due to failure of the SSCP-to-SSCP session—a double failure case. Judicious configuration of the network could ensure that most double failures still resulted in adequate notification.
62. During the assignment of sessions to virtual routes, a user exit is invoked. This exit can rotate the assignment of parallel sessions across several active virtual routes so that the failure of one virtual route will not disconnect the LUs, only reduce the maximum bandwidth between them. Other assignments can be made as needed by the application.
63. *NCCF General Information Manual*, GC27-0429; available through the local IBM Branch Office. This introduces the Network Communications Control Facility, which provides support for centralized network operator control as well as services used by IBM or customer-written communications network management applications.
64. *NPDA General Information Manual*, GC34-2010; available through the local IBM Branch Office. Describes the main purpose of the Network Problem Determination Application to be display of data at an NCCF operator's station.
65. H. L. Giles, "Successful network management hinges on control," *Data Communications* **7**, No. 8, 33-41 (August 1978).
66. Unpublished data assembled from running networks by Gerry Jacobs shows that approximately 85 percent of all calls to help desks are application-related. (This number may be smaller for mature, stable applications.)
67. J. D. Markov, M. W. Doss, and S. A. Mitchell, "A reliability model for data communications," *Conference Record ICC 78* **1**, 03.4.1-03.4.5, Toronto, Canada (1978).
68. SNA 4.2 only supports manual dial lines between subarea nodes.
69. In order for the SSCP restart, takeover, and normal domain activation sequences to be the same, the SSCPs must be able to resynchronize with running pieces of the network. The LUs that are actually in session have the definitive status of the LU-to-LU sessions; so, each LU tells its SSCP (on the response to ACTLU (ERP)) about its active sessions. Boundary functions provide this support for clusters and terminals.

Reprint Order No. G321-5096.