Interconnecting SNA networks

by J. H. Benjamin
M. L. Hess
R. A. Weingarten
W. R. Wheeler

Systems Network Architecture (SNA) allows terminals and application programs to communicate with one another using SNA entities called logical units. Until now, these logical units have had to be in the same network to communicate. This paper describes recently introduced SNA network interconnection functions that allow logical units in independent SNA networks to communicate with one another. Each network is configured, defined, and managed separately. By using one or more facilities called gateways, networks can remain independent while their logical units initiate, use, and terminate internetwork sessions, without any changes to themselves. A communications user need not be aware that a session partner is in a separate network.

Since the introduction of Systems Network Architecture (SNA)[1,2] in 1974, its functions have been continuously enhanced. At that time, SNA allowed terminals to be shared by data processing application programs in a single-host, tree-structured data communications network. In 1976, the networking capabilities were enhanced to allow programs in two or more hosts to communicate with one another and with the terminals. In 1979, parallel links and multiple routes among communication controllers and hosts were introduced to provide a full-mesh topology.[3]

Today, as more and more individual SNA networks are installed, there is a growing requirement for application programs in one SNA network to be accessible from terminals or application programs in another SNA network. When this is achieved, the networks are said to be *interconnected*. From a user's viewpoint, a set of interconnected networks is the same as a single network with an enlarged population of users. From a network manager's point of view, however, the autonomy of each individual SNA network is preserved.

The facility used to interconnect networks is generally called a *gateway*, a term that is consistent with common terminology to designate an entity that gives access to something. A gateway between networks accepts messages from one network, translates them to a form that can be understood in the other network, and transmits them to the appropriate destination. The amount and type of translation done by the gateway depend on how the protocols and physical media of the two networks differ.

This paper focuses on the problems of interconnecting networks with like protocols, specifically those specified by Systems Network Architecture. Other gateways have been defined for interconnecting similar networks. For example, the International Telephone and Telegraph Consultative Committee (CCITT) has defined an interface (X.75)[4] for interconnecting public data networks that offer an X.25 user interface. This approach uses a global addressing scheme (X.121) that is apparent at the user interface in each network. In another approach, the Advanced Research Projects Agency (ARPA) inter-

connects networks using the ARPA-specified virtual-call protocol between gateways, and routes datagrams from gateway to gateway as local network packets that include a global network address for the destination.[5,6]

The gateway that interconnects SNA networks is based on requirements and SNA concepts not found in these other examples of interconnection. This paper reviews basic SNA concepts and then describes the requirements that have motivated the gateway design. The components of the gateway and the protocols for using the gateway are described. Procedures for managing a multiple-network environment are also discussed. Finally, there is a description of an SNA network used as a case study during the gateway design.

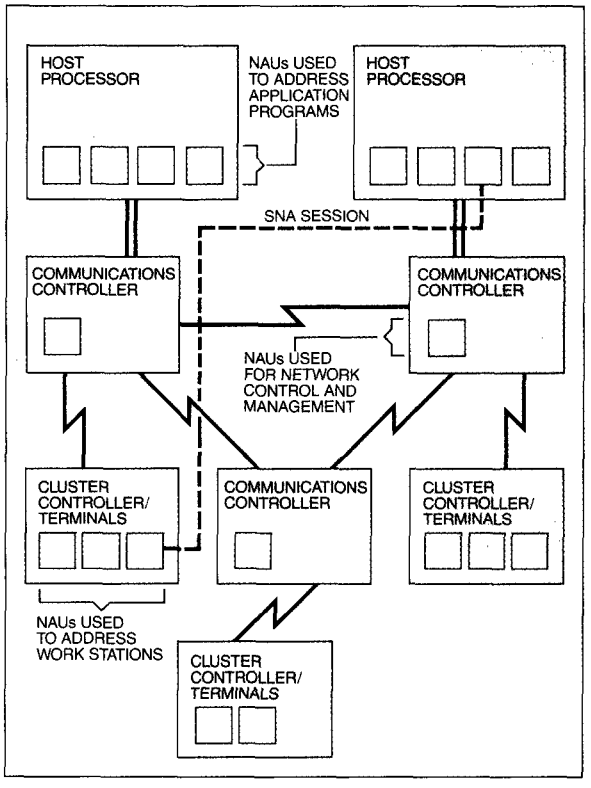
Characteristics of an SNA network

A network is a configuration of terminals, controllers, and processors, and the links that connect them. When such a configuration supports user applications involving data processing and information exchange, and conforms to the specifications of Systems Network Architecture, it is called an *SNA network*. Essentially, SNA defines logical entities that are related to the physical entities in a network and specifies the rules for interactions among those logical entities.

The logical entities of an SNA network include network addressable units and the path control network that connects them. Network addressable units communicate with one another using logical connections called *sessions*, as shown in Figure 1. The three types of *Network Addressable Units* (NAUs) are the Logical Unit (LU), the Physical Unit (PU), and the System Services Control Point (SSCP), which are defined as follows:

- *Logical Unit (LU).* An LU is a port through which end users may access the SNA network. An end user uses an LU to communicate with other end users and to request services of a System Services Control Point (SSCP).
- *Physical Unit (PU).* A PU is a component that manages the resources of a node in cooperation with an SSCP.
- *System Services Control Point (SSCP).* This is a focal point for configuration management, problem determination, and directory services for end users. SSCPs may have sessions with LUs and PUs. When such a session occurs, the LU or PU is in the *domain* of the SSCP. In addition to sessions with LUs and PUs, SSCPs may also communicate with each other to coordinate the initiation and termination of sessions between logical units in different domains.

Figure 1 Network Addressable Units (NAUs) and sessions in an SNA network



Each message sent to a network addressable unit is prefixed by a *transmission header*, which includes sixteen bits to represent the address of that network addressable unit. The address consists of two parts, the *subarea* field and the *element* field.

Each major node in the network is defined as a *subarea* node. Current subarea nodes are implemented either as hosts or as communication controllers.[7] The subarea field of the network address is used to route a message in the path control network from origin subarea to destination subarea, possibly through some intermediate subareas. The destination subarea node then delivers the message to the appropriate network addressable unit by using the element field of the address.

The number of bits (from 2 to 8) in the subarea field of a network address is selected by the network administrator. The remainder of the sixteen bits determines the maximum number of terminals or application programs in a subarea. This address-split in subarea and element fields must be the same for all nodes in the network to allow consistent network-wide routing. Because the choice of address-split is motivated by the characteristics of the network configuration, there is a high likelihood that two networks that want to interconnect will have different address-splits.

Any network-interconnection solution must maintain network independence and network management autonomy.

In addition to a network address, each network addressable unit has a *network name*. A network name is a symbolic identifier used to refer to a network addressable unit.

Thus, each network has a *name space* and an *address space*. An SSCP directory service consists of mapping names to addresses for those network addressable units in the SSCP's domain of control. A cooperative directory service is provided within each SSCP in a network to resolve names to addresses between domains, so that a directory entry at one SSCP points to the SSCP that can provide the resolution.

In an SNA network, when a session is set up between end users in different subareas, a particular physical path through the network's subareas and links is determined. The selection of this path is made indirectly through the specification of a *class-of-service* name. This symbolic name designates desired communication characteristics, such as path security, transmission priority, or bandwidth. The class-of-service name is mapped to a list of *virtual routes*, any one of which can be selected for the session. A virtual route is a logical connection between the two end users' subarea nodes. It supports protocols that provide data integrity, network transmission priority, and flow control functions.

A virtual route[3,8] is itself mapped to a set of links and nodes called an *explicit route*. The explicit route is the physical path that is used by the session. An explicit route may be shared by multiple virtual routes. In addition, each virtual route can be used by multiple sessions.

Required properties

The main requirement of any network-interconnection solution is to maintain network independence and network management autonomy while facilitating communication between the networks. This requirement has different facets, depending upon the perspective taken.

From an application programmer's or terminal operator's point of view, procedures for requesting and controlling SNA sessions should be the same regardless of whether the session partner is in the same or a different network. In this way, existing application programs and terminals would be able to initiate and participate in internetwork sessions using the same conventions, formats, and protocols that were used before network interconnection.

From the point of view of managing the configuration of an individual network, the act of interconnecting to another network should occur with minimum disruption. Here, minimum disruption means the following:

- There should not be a significant effect on a network's address space.
- It should not be necessary to understand which SSCP controls a logical unit in another network.
- It must be possible to have network names duplicated between interconnected networks.

Furthermore, once networks are interconnected, physical and logical configuration changes in one network should not necessitate corresponding changes in interconnected networks. For example, changing the address-split, route definitions, or domain definitions, or adding new links or nodes in one network should not cause corresponding definitional changes to the attached networks.

From a network-operations point of view, each network should be insulated so that malfunctions in

another network do not affect it. For example, one network should not be able to usurp control of another network's nodes or congest that network with data.

From a global configuration planning point of view, it should be possible to treat individual networks and their gateways as building blocks for the global configuration. Thus there should be sufficient flexibility to interconnect networks in tandem, to use parallel gateways between a pair of networks, or to interconnect multiple networks on a single gateway.

Although the independence requirement is paramount, it must be tempered with the pragmatics of network management. And though it should not be possible to seize control of another network's resources, it is useful to do fault isolation on an internetwork session. In addition, with appropriate authorizations, it should be possible to collect maintenance statistics and run tests on resources in other networks.

The solution is a gateway inserted between the networks to translate names and addresses.

Overview of the solution

Simply introducing a link to interconnect two SNA networks does not meet all these requirements. Most likely, the networks evolved with different address-splits and user-selected logical-unit names. For traffic to flow between the networks, route definitions are required. These would not be possible because of the ambiguities caused by duplicate addresses. The duplication of LU names makes session initiation requests ambiguous.

Figure 2 Network A's view of a gateway

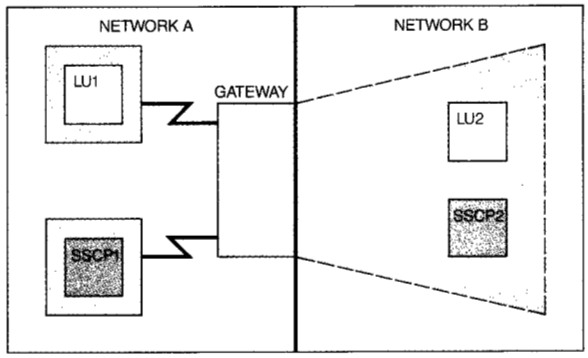
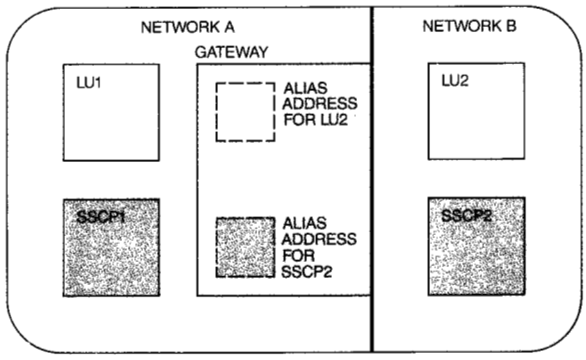


Figure 3 Gateway's view of network A's address space



The solution is a gateway inserted between the networks to translate names and addresses as necessary when messages are sent from one network to the other. Viewed from any one of the interconnecting networks, the gateway is a part of that network. In addition, the gateway's participation in other networks need not be known to a given network. For example, in Figure 2, the gateway is a subarea in network A. LU2 and SSCP2 are addressable in this subarea from network A. The fact that LU2 and SSCP2 are really in another network is not apparent to LU1.

Viewed from the gateway portrayed in Figure 3, the addresses in network A for LU2 and SSCP2 are alias addresses for network addressable units in network B. The gateway must translate addresses in network B to their alias addresses in network A. This translation is performed on the addresses in the transmission header of all messages sent on internetwork

Figure 4 Alias addresses used in internetwork sessions

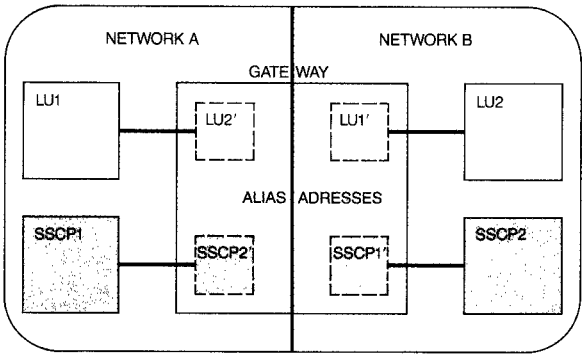
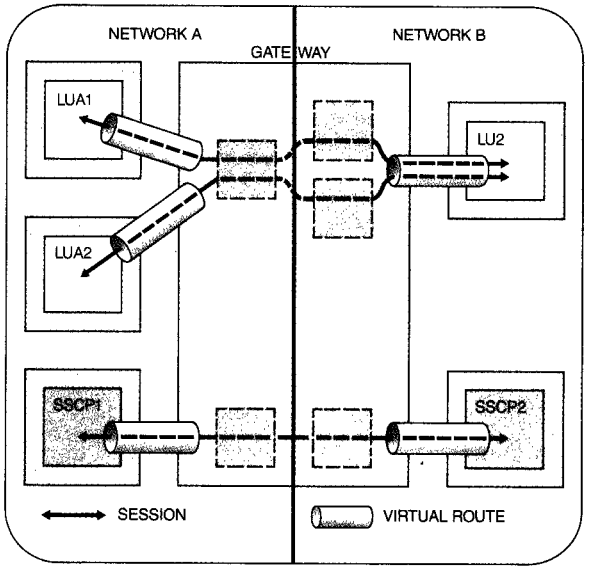


Figure 5 Virtual routes to a gateway



sessions. LUs and SSCPs in network A are also given alias addresses in the gateway to allow their representation to network B, as shown in Figure 4. PUs are not represented by alias addresses in the gateway, which prevents one network from taking control of another network's resources.

Session traffic from LU1 to LU2 will have different addresses in the transmission header fields, depending on which network it is currently traversing. In network A, the transmission header carries the real address of LU1 and the alias address representing LU2. When received by the gateway, the transmission header is translated to addresses understood in network B, which are the alias address representing LU1 and the real address of LU2.

The gateway serves as a virtual route end point. Routes are independently defined in each network. Figure 5 shows that the gateway will receive traffic from one virtual route and send it on another.

The use of alias addresses in the gateway to represent LUs and SSCPs in other networks limits the number of such resources that can be concurrently addressed. This limit is the number of element addresses in the gateway subarea. This limitation is alleviated through dynamic use of the element addresses in the gateway subarea. Later it will be shown how, at session initiation, a free address in the gateway is dynamically assigned to represent the LU or SSCP in another network. At session termination, when there is no further need for the address, it is returned to the free pool so that it may subsequently represent a different network addressable unit. Thus, the number of successively addressable network addressable units is not limited. The apparent number of terminals or application programs accessible to each end user increases above what is normally available within one SNA network.

The use of alias addresses allows communication with network addressable units in other networks.

The use of alias addresses allows communication with network addressable units in other networks as though they were part of the same network. Similarly, alias names can be used to represent other network names to avoid ambiguities that would otherwise result from duplicate names in the networks. LU names play a significant role in session initiation protocols. Figure 6 shows that the LU known as CICS[9] in network B can be assigned a different name (BIGCICS) in network A to avoid

ambiguity with the already existing CICS in network A. The gateway is responsible for translating the alias names to their real counterparts. The details of name translation are covered in a later section.

Alternative approaches

The address translations done by the gateway are essentially at the path control layer of SNA protocol. By intervening in the path control protocols, the gateway does not affect the session protocols. In the selected solution, the full session identifiers (origin address, destination address) are translated from network to network, giving a session endpoint the illusion that the session is local to the network of that endpoint.

Other levels of interconnection[10] that were considered but not selected were those of the logical unit level[11] and another path control alternative at the explicit route level.[12]

Intervention at the path control level and transformation of full network addresses are less wasteful of a network's address space, compared with the other two alternatives. In any network, a single subarea address in the gateway provides enough element addresses to represent logical units in many subareas in other networks. A disadvantage of this approach is that there is no way to translate virtual route identifiers transparently. The gateway must become a virtual route endpoint, and, therefore, an end-to-end virtual route does not exist. The virtual route flow control mechanism cannot as effectively protect the gateway from congestion. On the positive side, by interrupting the virtual route, virtual route protocol incompatibilities between networks are insulated by the gateway. This offers an additional level of flow control protection to networks that support virtual routes when attaching networks that do not support virtual routes.

Gateway components

The gateway has a System Services Control Point (SSCP) that is enhanced to reroute session services requests between SSCPs in separate networks and to make the appropriate name and address translations within those requests. This component is called the gateway SSCP.[13] In addition, the gateway has a component that allows it to appear to be a subarea in each network to which it attaches. This component, called the gateway node,[14] translates the network addresses in the transmission headers of messages sent between networks. The functional responsibilities of the two components are shown in Figure 7.[15]

Since session traffic between logical units does not pass through the gateway SSCP, it incurs very little overhead in traversing network boundaries. The less frequent SSCP-to-SSCP traffic requires the services of both the gateway SSCP and the gateway node. At session initiation time, the dynamic assignment of alias addresses in the gateway node is done cooperatively by the gateway SSCP and the gateway node.

Figure 6 Alias names to avoid ambiguity

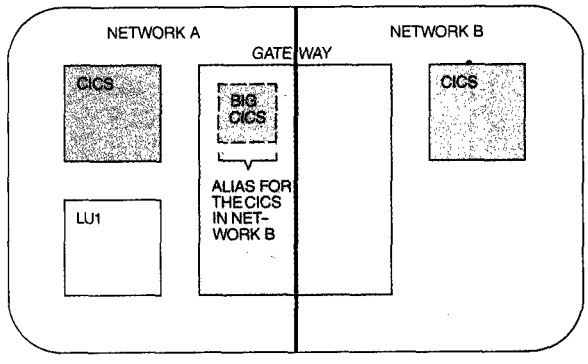


Figure 7 Function distribution in a gateway

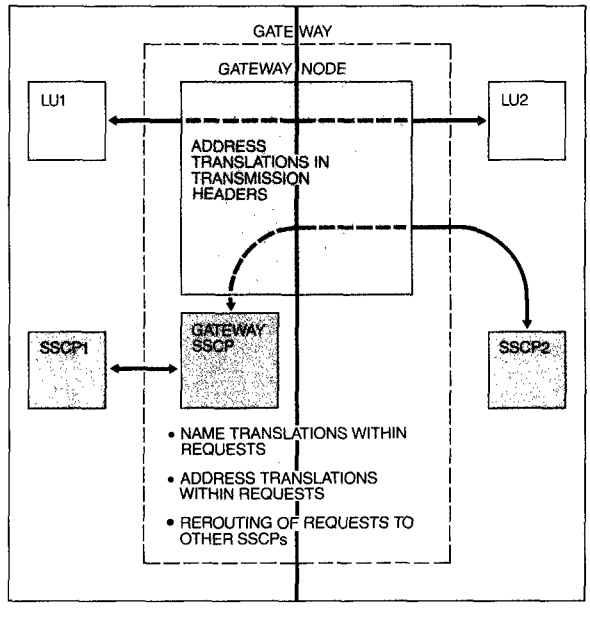
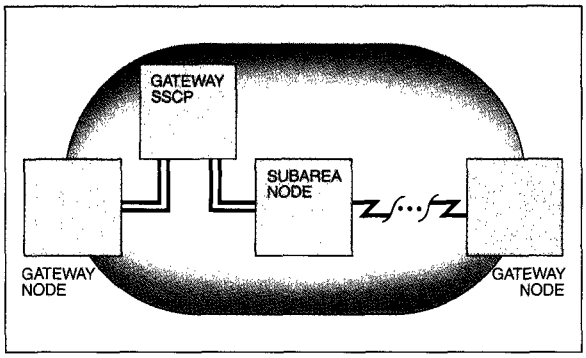


Figure 8 Gateway SSCP connections to gateway nodes



The physical components of a gateway are a host computer for the gateway SSCP and a communications controller for the gateway node. Figure 8 shows that the gateway SSCP can connect to a gateway node directly by a channel or indirectly by telecommunication links through an SNA network. The gateway SSCP might also be the SSCP of one of the logical units for an internetwork session. The following discussion focuses on the general case, where neither logical unit is in the domain of the gateway SSCP.

Logical considerations when creating a gateway

In addition to establishing the physical connections between the gateway components and between the gateway node and an attached-network node, creating a gateway has some logical considerations.

Provision must be made for assigning network identifiers. Each network that connects to other SNA networks is given a different symbolic identifier. These network identifiers are used by the gateway node and the gateway host to indicate the network to which a particular name or address applies. The identifiers are used as qualifiers of names and addresses in the requests exchanged by gateway components to initiate or terminate sessions. End users and logical units joined in sessions, however, need not be aware of network identifiers. Furthermore, the path control network routes data on sessions without using network identifiers.

To support communication between two logical units in separate SNA networks, each network must have a way to refer to the logical unit in the other network that, at the same time, avoids LU name conflicts. This is primarily so that an end user of a logical unit in one network can request a session with the logical unit in the other network. The obvious choice is to refer to an LU by the network name assigned in its own network, but that might match a resource name in the requesting network's name space, resulting in ambiguity.

Logical units can be renamed to eliminate name conflicts, but this would require the regeneration of network control programs or tables in host access methods. The effect of this might extend to application programs, network operators, and terminal users. Renaming logical units violates the objective of allowing interconnected networks to have independent name spaces.

The alternative of always qualifying a logical-unit name with a network identifier was rejected because it violates the objective of making internetwork connections transparent to end users and existing SNA access methods.

The SNA interconnection function provides an alternative that meets the objectives. Within each network, alias names can be used to refer to logical units that are actually in other networks. Each alias name is user-assigned to meet the constraints of the network where it is used. A name translation function is installed at the gateway host, and user-defined tables are created that map the alias logical-unit names to the actual logical-unit names and corresponding network identifiers. The gateway SSCP uses the name translation function when processing a request for a session with a logical unit designated by an alias name.

A request for a session with a logical unit in another network must always be directed to the gateway SSCP, whether an alias or real name is used for that logical unit. This is easily accomplished at any SSCP that wants its logical units to request sessions with the other-network LU. The name of the logical unit is added to the table previously used only as a directory for logical units in other domains of the same network, and the gateway SSCP is named as its control point. Defining other-network logical units this way gives existing SNA products access to those logical units without upgrading the products to be aware of the gateway.

At the gateway SSCP, there are tables to direct the session setup request to the correct SSCP. The

combination of network identifier and SSCP name is used as a unique identifier of the control point for the requested LU.

The gateway SSCP is an intermediary between SSCPs in the interconnected networks. As an intermediary, it has sessions with SSCPs in each network. Using these SSCP-to-SSCP sessions, the gateway

Design for the gateway evolved from previous multiple-domain session protocols in SNA.

SSCP transfers LU-to-LU session initiation requests from one network to the other. The successive SSCP sessions act as a setup path for the LU sessions.

Tables used previously by SNA access methods are still used to support SSCP-to-SSCP sessions. These tables specify the name and address of each SSCP that may have a session with a particular SSCP. Existing hosts do not have to upgrade their access methods to establish a session with the gateway control point.

Some new definitions are used at the gateway host to specify SSCPs in other networks. These allow the gateway SSCP, once it determines the network identifier and SSCP name of the control point for a logical unit, to send a session setup request on the appropriate setup path. In a simple configuration with one gateway host and one gateway node, the gateway SSCP must have an active session with the SSCP that controls the requested logical unit. For some of the more complex configurations described later, the setup path can go through other gateway SSCPs on the way to the correct SSCP.

Each attached network views the gateway node as a set of network addressable units in a particular subarea of that network's address space. Routing tables in the subarea nodes of each network include the gateway node's subarea address as a destination subarea. Definition statements for the gateway node reflect this view by specifying the network identifier of each attached network, the subarea address of the gateway node in that network, and the address-split in that network. In addition, the gateway node contains routing tables for each network and is told what network is connected by each physical link leading to other subarea nodes.

For each network, a subset of the network addresses that it perceives in the gateway node is set aside as a pool of addresses to represent network addressable units that are actually in other networks.

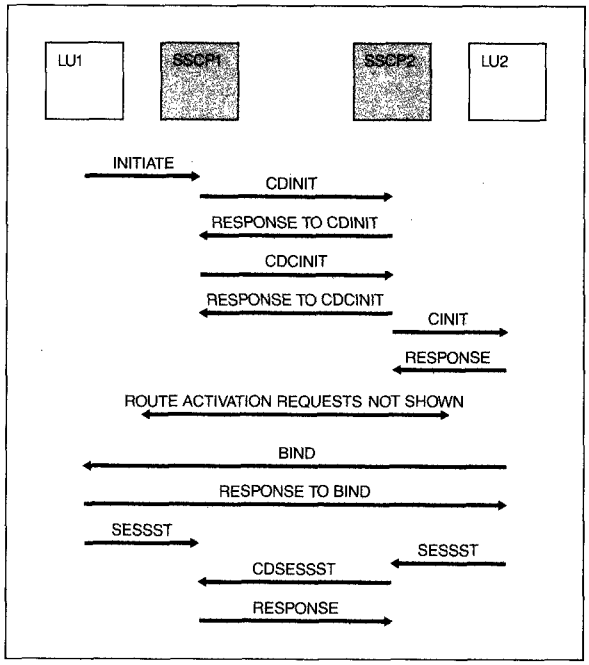
Establishing internetwork sessions

Design for the gateway evolved from previous multiple-domain session protocols in SNA. The gateway is essentially a translator inserted on the path between network addressable units in different domains to account for the domains' being in independent networks. All session services requests for LU-to-LU sessions that can be exchanged between two SSCPs in separate domains can now be directed to the gateway SSCP and rerouted to the destination SSCP in another network. Before the requests leave the gateway, names and addresses are translated so that they are understood in the destination network. A control point that is not a gateway SSCP uses normal cross-domain protocols with the gateway SSCP and is unaware of internetwork activity within or beyond the gateway. Each logical unit involved in an internetwork session need not be aware that other networks are involved.

Thus it is instructive to view the gateway SSCP as a translator inserted on the session setup and takedown path of a cross-domain session. Similarly, the gateway node can be viewed as a translator inserted on the session data path. A brief review of cross-domain session activation sequences follows, after which the internetwork session activation sequence is described. (New terms introduced in this section in full capitals, such as BIND and INITIATE, refer to particular request or response messages, simply called requests or responses.)

An active session between two logical units in separate domains of a network is established after an exchange of session initiation requests between the two SSCPs that control the logical units, and an exchange of BIND SESSION request and response between the logical units themselves. A BIND SESSION is a request that flows between logical units to activate a session between the logical units. The term BIND is usually used alone. To exchange

Figure 9 Establishing a cross-domain session



session initiation requests, the cooperating SSCPs must be in session. Two SSCPs send special messages to set up their own session.

There are several variations of LU-to-LU session initiation, such as by either LU, by a third-party LU, and by the network operator. The initiation request indicates which LU is to be the primary LU, or sender of BIND, and which LU is to be the secondary LU, or receiver of BIND. Class-of-service names may be specified, or they can be derived from user-defined tables. One representative variation is developed as a reference for comparison with internetwork protocols.

Figure 9 shows how a secondary logical unit LU1 in the domain of SSCP1 initiates a session with LU2 in the domain of SSCP2. Such a sequence is commonly triggered by a logon request from the end user of LU1. SSCP1 receives an INITIATE request from LU1, asking for a session with LU2 and containing a mode name, which is a symbolic reference to the set of rules to be used for the session. From user-defined tables, SSCP1 determines that LU2 is in the domain of SSCP2. A CROSS-DOMAIN INITIATE (CDINIT) request is sent to SSCP2. The CDINIT includes the name and address of LU1, the mode name, the class-of-service name derived from the mode name, and the name of LU2. SSCP2 returns the address of LU2 in the CDINIT response. As the SSCP of the secondary LU, SSCP1 resolves the mode name to session parameters and sends the session parameters to the SSCP of the primary LU in a CROSS-DOMAIN CONTROL INITIATE (CDCINIT) request. As the SSCP of the primary LU, SSCP2 resolves the class-of-service name to a list of virtual routes, which is passed to LU2 along with the session parameters in the CONTROL INITIATE (CINIT) request. BIND is sent from LU2 to LU1 on the first route in the list that can be activated. After the positive response to BIND, the LU-to-LU session is active. The two SSCPs are informed of the active session by the SESSION STARTED (SESSST) and CROSS-DOMAIN SESSION STARTED (CDSESSST) requests. This is shown in Figure 9.

Neither LU1 nor LU2 is aware that its session partner is in another domain. The two SSCPs appear as a single SSCP to each LU, and neither is aware of the requests exchanged within this *composite SSCP*. Compare Figure 10 with Figure 11 to see that the view of each LU is the same, regardless of whether the other LU is in the same or a separate domain.

Assume now that LU1 and SSCP1 are in one network, identified as NETA, and that LU2 and SSCP2 are in another network, identified as NETB. Pursuing the view that the gateway is inserted on the path between different domains to account for the domains being in different networks, the composite SSCP of Figure 11 now includes the gateway SSCP. This is shown in Figure 12.

In our example, the gateway SSCP is in NETA, and has a same-network session with SSCP1 and an internetwork session with SSCP2. The gateway node is inserted on the session path between the gateway SSCP and SSCP2, as well as on the session path between LU1 and LU2. Each LU still perceives a single SSCP and is not aware of requests exchanged within the composite SSCP. Both SSCP1 and SSCP2 use normal cross-domain protocols with the gateway SSCP. They need not be aware of the requests sent within the gateway and beyond the gateway. The gateway SSCP sends new requests to the gateway node to set up name and address transforms for internetwork sessions.

Given the prerequisite SSCP-to-SSCP sessions, LU1 in NETA can set up a session with LU2 in NETB. The

sequence of requests is shown in Figure 13. Because NETA already contains a logical unit called LU2, the INITIATE request uses the alias name LUX to refer to the LU2 that is in NETB. This does not mean that the user of LU1 must change session initiation procedures or be aware of the location of the requested logical unit. It means only that the user of LU1 knows there is a logical unit called LUX that provides the needed services.

Just as for a cross-domain session, SSCP1 sends a CROSS-DOMAIN INITIATE (CDINIT) request to the SSCP specified in its tables as the owner of LUX, which is the gateway SSCP in our example. The CDINIT includes the mode name, the class-of-service name, the name of LU1, the address of LU1, and the name LUX. Since the gateway SSCP has no definition of LUX, it uses the name translation function and determines that LUX corresponds to LU2 in NETB owned by SSCP2. Using the destination network identifier, the name translation tables are also searched to find the alias name used in NETB for LU1, the class-of-service name in NETB, and the mode name in NETB. All of these are returned to the gateway SSCP.

Now that it knows the requested logical unit is in another network, the gateway SSCP sends a REQUEST NETWORK ADDRESS ASSIGNMENT (RNAA) asking the gateway node to allocate a pair of alias addresses for the session. The network identifiers and names of the logical units are included in the RNAA, along with the network

Figure 10 Session initiation requests within one domain

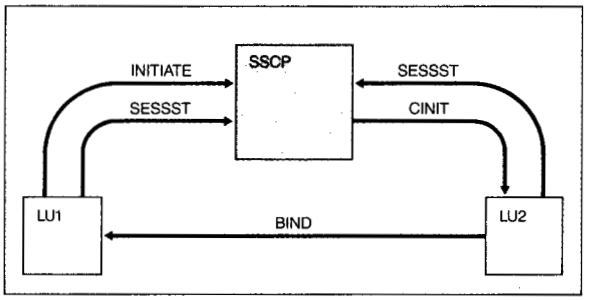


Figure 11 Session initiation requests between two domains

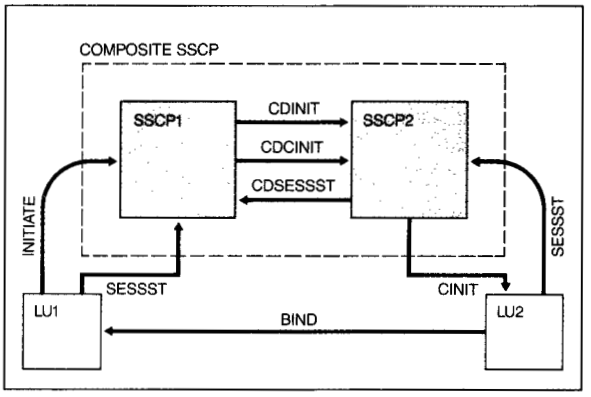


Figure 12 Session initiation requests between two networks

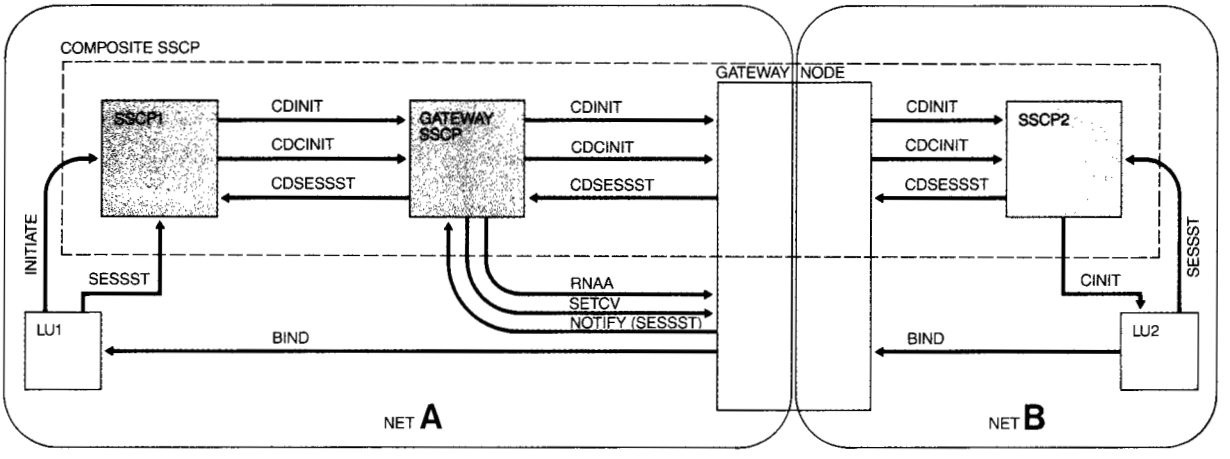
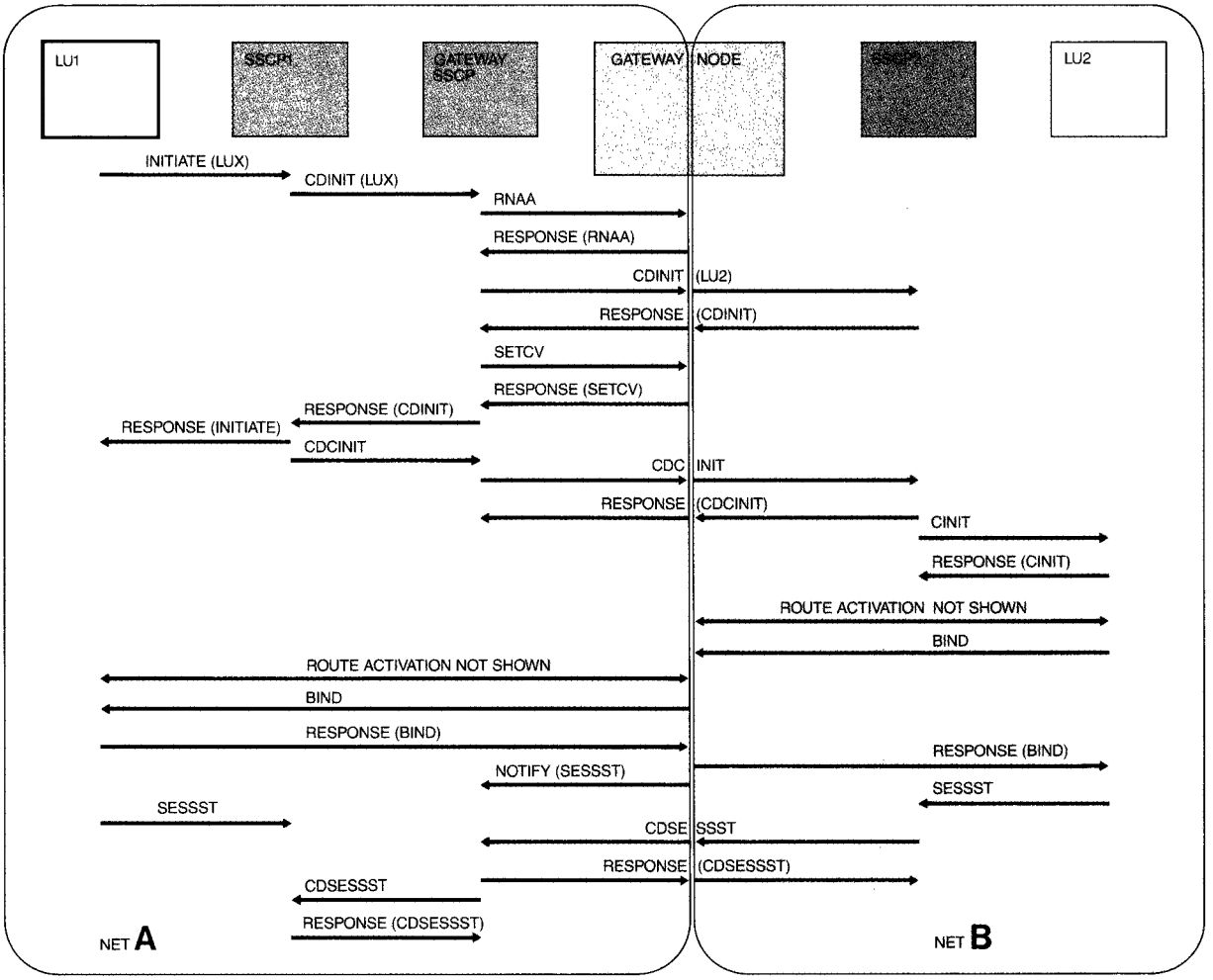


Figure 13 Establishing an internetwork session



address of LU1. The gateway node assigns an address in NETA to represent LU2 and an address in NETB to represent LU1. These alias addresses are returned in the response to RNAA.

Before sending CDINIT to SSCP2, the gateway SSCP changes the name fields to carry the mode, class-of-service, origin LU, and destination LU names understood in the name space of NETB. The origin LU address is changed to the alias address assigned for LU1 in the NETB subarea of the gateway node. As CDINIT passes through the gateway node, only the transmission header is changed to represent the gateway SSCP-to-SSCP2 session in NETB. SSCP2 processes the CDINIT as the owner of LU2 and returns the address of LU2 in the response.

After receiving the response to CDINIT, the gateway SSCP has all the information required to complete the transforms in the gateway node. The network address of LU2 in NETB from the response completes the address mapping started with RNAA. Both the alias and real names of the two logical units are known from CDINIT and the name translation tables. And the class-of-service name in NETA resolves to the list of virtual routes for the session path between the gateway node and LU1. This list is needed when BIND from the primary LU arrives at the gateway node. A SET CONTROL VECTOR (SETCV) request gives the address of LU2, the alias LU names, and the virtual route list to the gateway node. All of this information is used by the gateway node when it receives BIND.

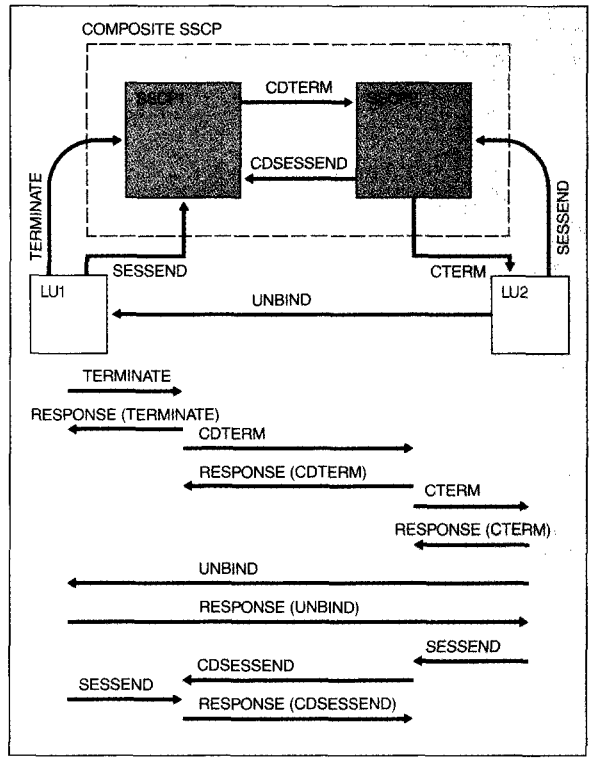
Before forwarding the CDINIT response to SSCP1, the gateway SSCP changes the destination LU address field to carry the alias address for LU2 in the NETA subarea of the gateway node. The class-of-service name field carries the name that applies in NETA. SSCP1 handles the response as for a cross-domain session. It sends a response to LU1 for the original INITIATE, resolves the mode name to session parameters, and sends the session parameters to the other SSCP in CROSS-DOMAIN CONTROL INITIATE (CDCINIT).

Modifications to the flow control protocol help the gateway node protect against one network monopolizing its buffers.

Except for BIND processing, the rest of the session setup sequence is just a matter of rerouting requests in the gateway SSCP after translating name and address fields, as needed, and changing transmission headers in the gateway node. When BIND arrives at the gateway node, the virtual route list sent on the SETCV is used to select and activate a route from the gateway node to the subarea node for LU1. The same requests used by a host node to activate routes are used by the gateway node. Before BIND is sent on to LU1, the logical unit name fields within the BIND are changed to carry the names understood by LU1. The primary LU name is changed from LU2 to LUX and the secondary LU name is changed from the alias used in NETB to LU1. If a negotiable BIND is used to allow the secondary LU to return suggested BIND parameters to the primary LU, the gateway node does the reverse translation of LU names in the BIND response.

When transferring data on internetwork sessions, neither LU1 nor LU2 realizes that the other is in a separate network. The SNA protocols used by the logical units to exchange data on the session are unchanged. Network addresses in the transmission headers of messages on the session are translated when the messages cross a network boundary, but the session partners are not aware of this.

Figure 14 Session termination requests between two domains



The path of the session includes routes in each of the networks traversed, with route endpoints in the gateway node. Some modifications to the flow control protocol that operates at virtual route endpoints help the gateway node protect against one network monopolizing its buffers. The gateway node maintains counters for each virtual route endpoint. When a message is received at a virtual route endpoint, the counter for that route is incremented. That counter is decremented when the message leaves the gateway node. If the count reaches a threshold value, the gateway node withholds virtual route pacing responses[16] for the congested route. This restricts the flow on that virtual route until the congestion has subsided.

Terminating internetwork sessions

Normal termination of an internetwork session is, to the logical units and their SSCPs, the same as ending

a cross-domain session. Figure 14 shows the way in which either the primary LU or the secondary LU can end a cross-domain session. If the secondary LU requests termination, the sequence starts with a TERMINATE (TERM) request from LU1 to SSCP1. If the primary LU terminates the session, the sequence starts with an UNBIND request from LU2 to LU1. Each LU still sees the composite SSCP as one SSCP.

When terminating an internetwork LU-to-LU session, this view of the composite SSCP does not change, even though each LU and its owning SSCP are in different networks. Figure 15 shows how the gateway SSCP reroutes CROSS-DOMAIN TERMINATE (CDTERM) and CROSS-DOMAIN SESSION ENDED (CDSESSEND) requests within the composite SSCP shown. Except for UNBIND processing, the sequence is just a matter of rerouting requests in the gateway SSCP after translating name and address fields as needed, and changing transmission headers in the gateway node.

Before forwarding the response to UNBIND, the gateway node discards its record of the session.

Figure 15 Terminating an internetwork session

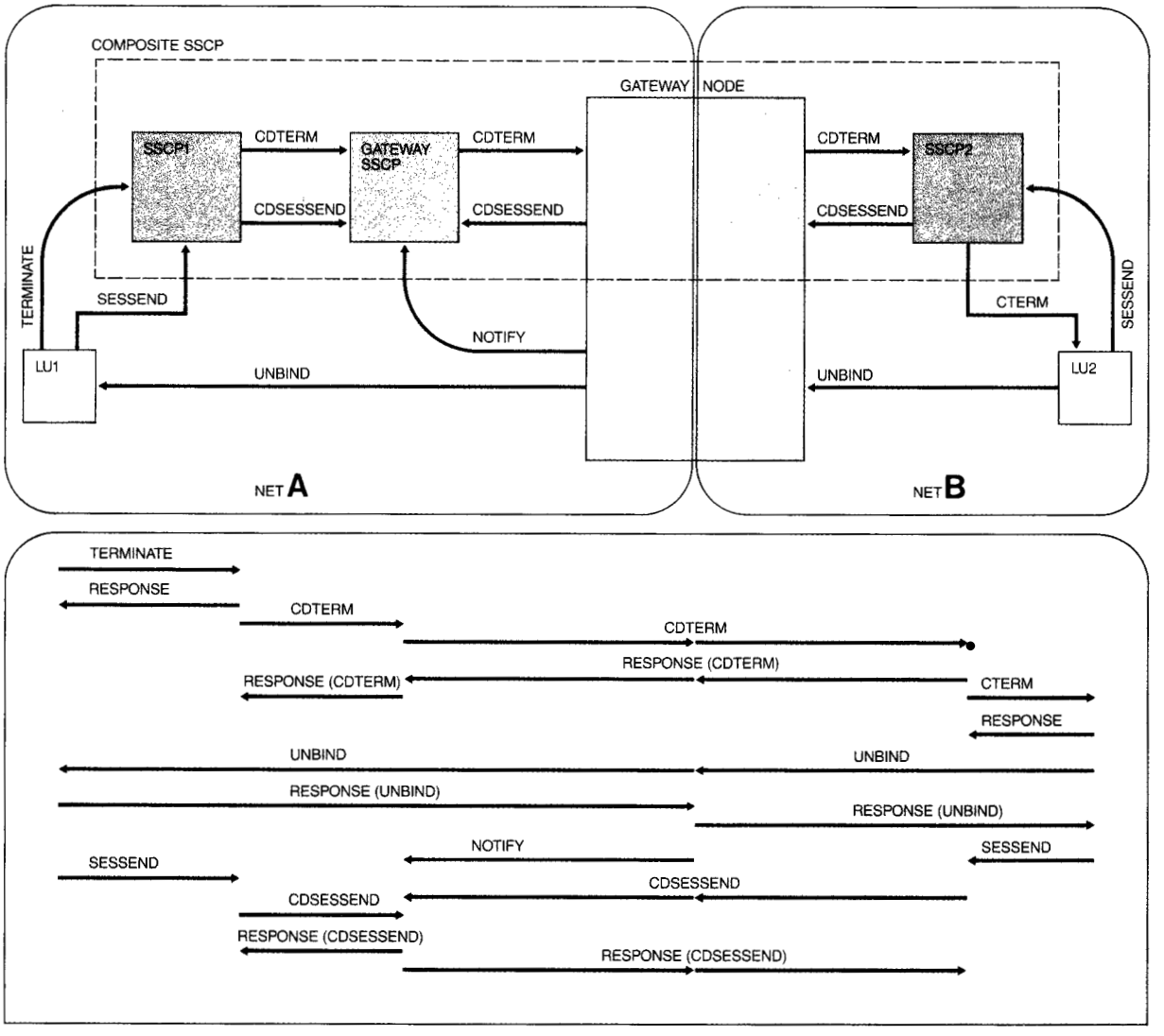
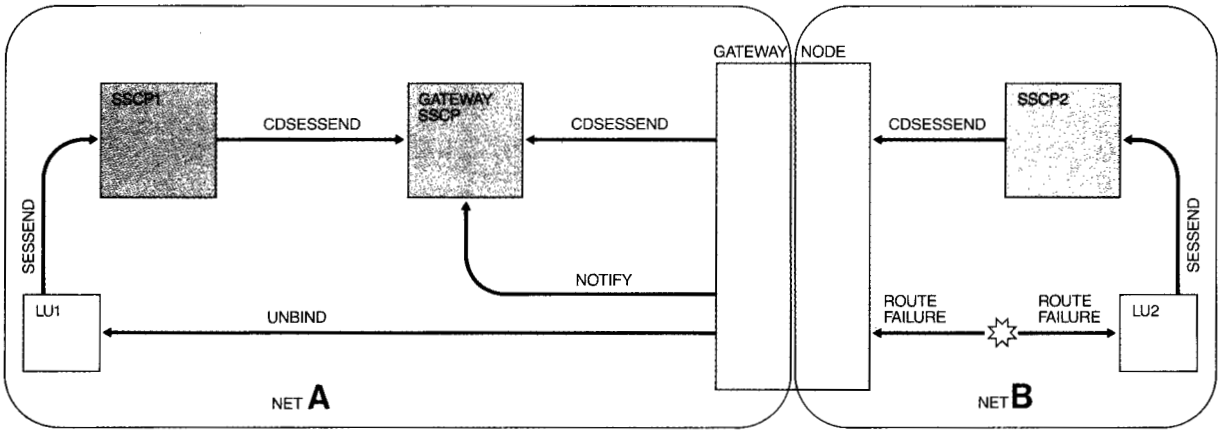


Figure 16 Outage notification for an internetwork session



Each dynamically assigned alias address is returned to the pool for its network, unless it is supporting other sessions with the same LU. To be sure the gateway SSCP knows the internetwork session has ended, the gateway node sends it a NOTIFY request. This notifies the gateway SSCP that it is safe to discard its record of the session, even if the CDSESSEND requests do not reach the gateway SSCP because of network failures. Failures on any of the session paths between the SSCPs or the LUs and their SSCPs might prevent CDSESSEND from reaching the gateway SSCP.

If a node or link failure in a network prevents data from being sent, the SNA nodes that detect the failure make sure the session partners affected are notified. For example, when a logical connection between two subareas fails, each of the detecting subareas originates a "route failure" message. Each failure message is propagated from subarea node to subarea node until it reaches an end node of the affected route. In the end node, each session using the inoperative route is deactivated by a session deactivation request, which appears to be from the session partner. This prevents deadlocks and allows the session to be reinitiated on a different route.

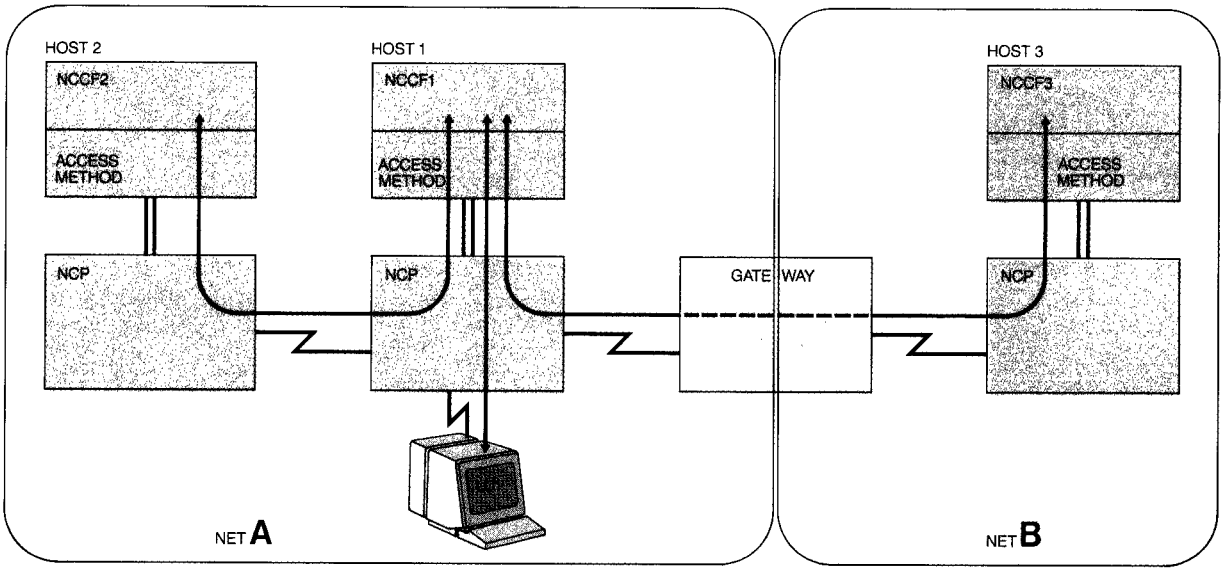
To extend this technique to interconnected networks, the gateway node sends a session deactivation request along the path of an internetwork session, when the route used by the session in one of the attached networks fails. Figure 16 depicts this. Failure of the route between the gateway node and LU2 is reported to the gateway node with the "route failure" message. Using its record of internetwork sessions, the gateway node determines that the session between LU1 and LU2 is on the failing route. UNBIND is sent from the gateway node to LU1 as though it came from LU2. The response to UNBIND causes the gateway node to send NOTIFY to the gateway SSCP, as for normal session termination.

Controlling and managing interconnected networks

Network operators in SNA networks, whether terminal users or operator programs, issue commands that ultimately are processed by SSCPs. The SSCPs interpret the commands and do the requested actions, such as activating or deactivating resources, starting or stopping traces, and displaying status information. Each access method that contains an SSCP defines the syntax and meanings of the operator commands it provides. The objective for interconnected networks is that existing operator commands processed at any host provide the same functions with respect to that host's network as they would if issued in a single-network environment.

Several existing commands specify the name of a logical unit in another domain and do such things as allow cross-domain sessions or terminate existing cross-domain sessions with that logical unit. The same commands can be used to allow or terminate sessions with a logical unit in another network. The operator need not be aware of the other network because the name space of any domain whose logical units initiate or accept sessions with a logical unit in another network includes a name for that

Figure 17 NCCF sessions with interconnected networks



logical unit. This name, which might be an alias, can be specified on operator commands.

There are a few instances where a network operator needs to know about other networks just to avoid confusion. Some messages to the operator from the gateway SSCP include network identifiers to indicate the networks that contain the resources named in the messages. An optional network identifier may be specified on some commands requesting status of resources from the gateway SSCP. The network identifier limits the scope of such a command to representations of resources in that network.

Operator commands processed by a control point of an access method in one network do not control the resources in another network. This isolation of one network from another is a fundamental objective of interconnecting networks rather than integrating them. Those who want an operator in one network to control resources in an interconnected network, however, can use the Network Communications Control Facility (NCCF) program product provided by IBM.

NCCF allows an operator at a terminal in one domain of an SNA network to issue operator-control commands to access methods in other domains of the same or a different network.[17] This capability is illustrated in Figure 17. Terminal LU1 has a session with NCCF1, and NCCF1 has a session with NCCF2 in NETA and NCCF3 in NETB. The NCCF operator can enter a command at terminal LU1 and cause NCCF1 to route the command to either NCCF2 or NCCF3. Each NCCF presents the command to the access method in its host. Command responses are returned to LU1 on the reverse path. Thus, control of multiple networks can be centralized at one operator. Another possibility is to give one operator control of resources related to the gateway and to distribute control of other resources to other operators.

Problem determination

Facilities are available in each network to determine the cause of problems within that network. Sessions through a gateway, however, complicate problem determination because the session path leaves the jurisdiction of one network and enters the jurisdiction of another. The change of network address space that occurs when a message goes into another network—thereby causing changes to the transmission header—also complicates problem determination. To deal with this additional complexity, existing problem-determination tools are enhanced for multiple-network configurations.

The subarea nodes in a network in conjunction with the Network Logical Data Manager (NLDM)[18]—an

NCCF-based communications network management application program—collect information for diagnosing session-related problems. Each session is recorded in the NLDM data base, along with appropriate trace data and configuration data for the session path. Several hosts, including the gateway host, may have data for an internetwork session. NLDM communication to another NLDM, via appropriate NCCF-to-NCCF sessions, is used to gather all the data related to a particular session from the various hosts, so that a composite view can be presented at one of the hosts. The paper in this issue by Weingarten and Iacobucci[19] gives details on NLDM in both single- and multiple-network environments.

Configurations

The simplest gateway to interconnect two SNA networks consists of one gateway SSCP and a gateway node, as described in the previous examples. This configuration is sufficient to demonstrate the main concepts and protocols for interconnecting SNA networks. Other configurations that meet additional user requirements are possible.

Several networks can be interconnected with just the simple gateway. Logical units in any two of the networks can have active sessions, regardless of which network contains the gateway SSCP. For example, Figure 18 shows four networks interconnected using the simple gateway. Here the data base application in NETD has sessions with LU1 in NETA, LU2 in NETB, and LU3 in NETC. Assuming that LU1, LU2, and LU3 initiate the sessions with the application, their initiation requests first go to their SSCPs, which send cross-domain requests to the gateway SSCP. These trigger the gateway SSCP to send initiation requests to the SSCP for the application.

Interconnecting multiple networks with one simple gateway avoids costs for some of the networks.

Interconnecting multiple networks with one simple gateway avoids costs for some of the networks.

Figure 18 Simple gateway interconnecting multiple networks

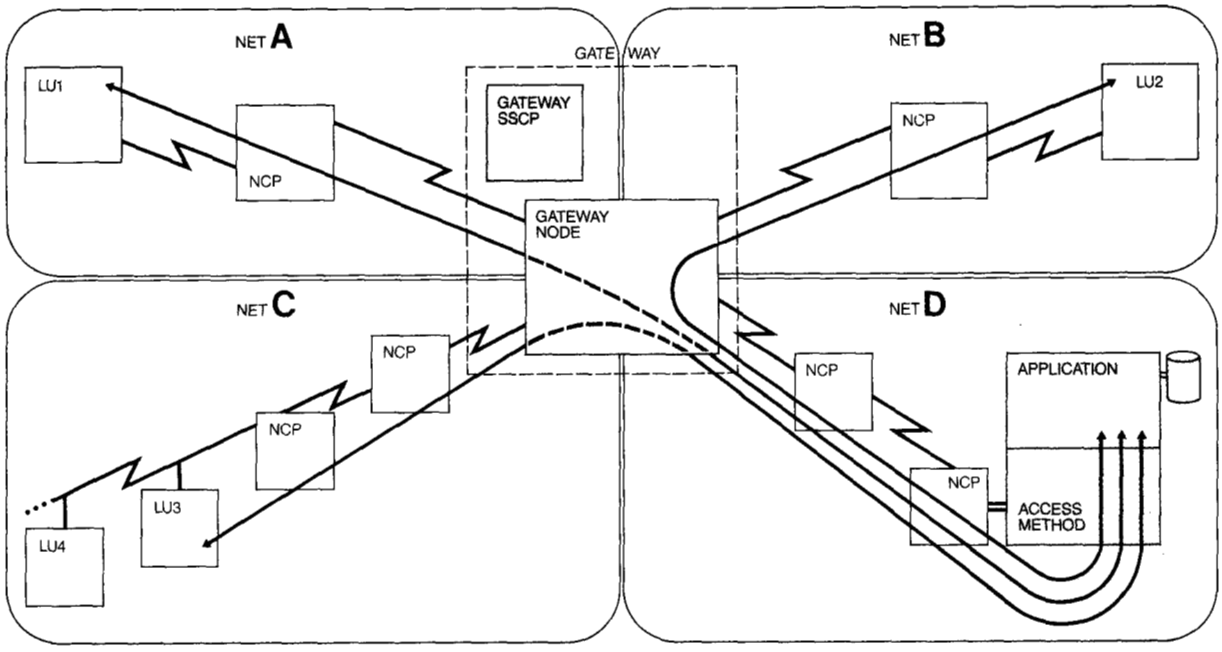
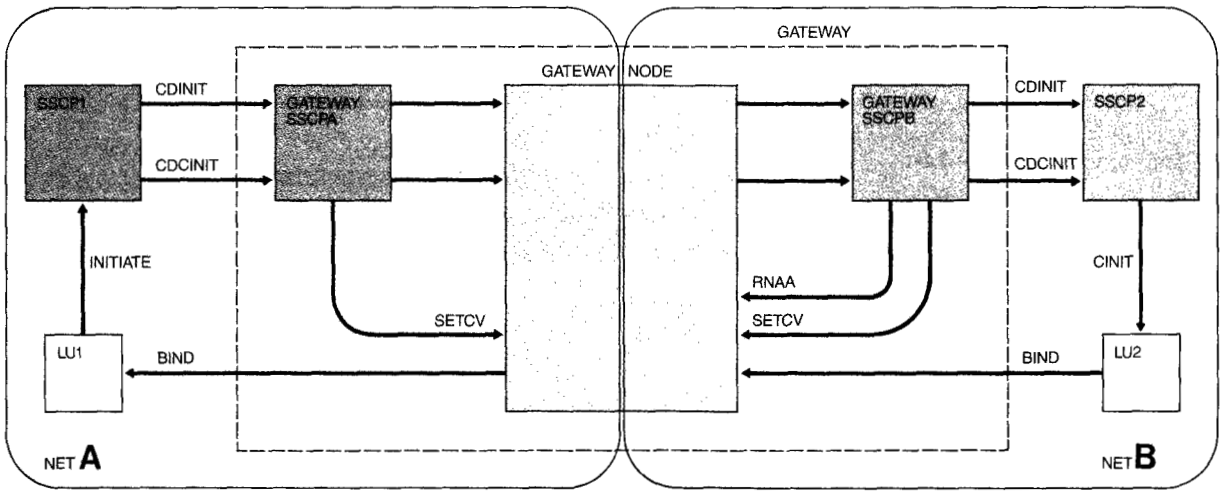


Figure 19 Shared-control gateway



Rather than install their own gateway nodes, gateway hosts, and name translation tables, these networks share the gateway. By sharing the gateway, the networks also limit the number of sites where people must be present who know how to install and manage a gateway.

Some enterprises have several systems with SNA access methods distributed throughout the enterprise, each controlling its own small network. For example, there might be multiple IBM 4331 processors using the Advanced Communications Function for VTAM Entry (ACF/VTAME) program product to control communication between the host processor and terminals attached to the processor through a communication adapter. At times, these systems can benefit from communicating with an existing enterprise-wide network. Integrating all the systems into the existing network takes considerable system definition effort. Instead, each system can be treated as a separate network and can be attached to a simple gateway controlled by a gateway SSCP in the enterprise-wide network.

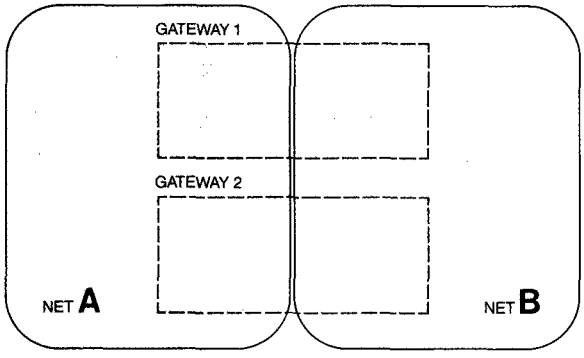
When two networks interconnect, each network owner might install a gateway SSCP and thereby achieve a shared-control gateway. One reason for doing this is to divide the tables that relate names of logical units to the names of owning SSCPs, such that the table at each gateway SSCP is only for logical units in its own network. This option confines the responsibility of knowing the actual location of a logical unit to the network that contains the logical unit. When logical units are moved within one of the networks, only the tables within that network are changed.

To see how this is done, refer to Figure 19. The gateway SSCPs have a session with each other, and each has a session with the gateway node, making them part of the same gateway. As in the example of setting up a session through a simple gateway, SSCP1 sends a CDINIT request to gateway SSCPA when LU1 initiates a session with LU2. Gateway SSCPA has a special table that directs session setup requests for any logical units in NETB to gateway SSCPB. Gateway SSCPB has the normal table of logical units in other domains of its network, and SSCP2 is identified as the owner of LU2. Since SSCPB is a gateway SSCP, it can reroute CDINIT to SSCP2.

Unless defined otherwise, gateway SSCPs automatically share control of a common gateway node as shown in Figure 19. These defaults are based on which SSCP naturally has the information needed to create the requests sent to the gateway node. They also allow flexibility in installing name translation support. If name translations are needed, any one or all of the gateway SSCPs within the gateway can have name translation tables.

Shared-control gateways have other advantages besides isolating each network from the actual location of logical units in other networks. The

Figure 20 Parallel gateways between two networks



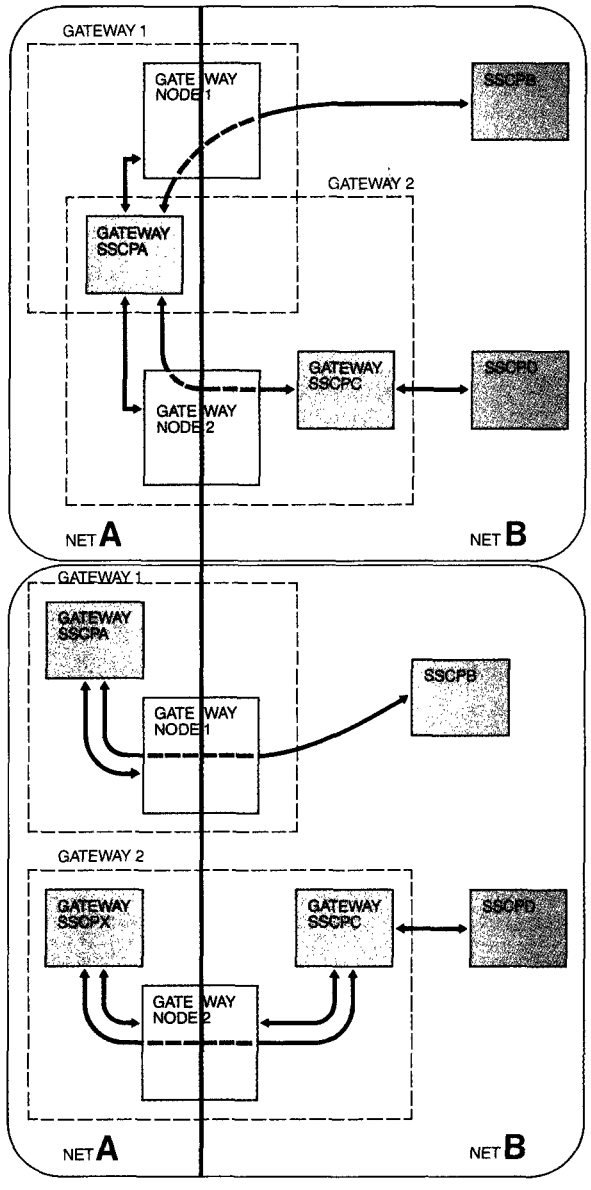
gateway SSCPs receive complete knowledge of the internetwork sessions through the shared gateway node. Each gateway SSCP can do authorization checking on use of the gateway. Instead of multiple SSCP-to-SSCP sessions through the gateway, there is one that simplifies routing of internetwork requests and leaves more alias addresses available for LU-to-LU sessions.

In some instances, it is not appropriate for a network to share control of a gateway node with another network, even if both contain gateway SSCPs. For example, a service bureau that provides application programs or data transport services to other networks by attaching the networks to a gateway node might take sole responsibility for controlling the gateway node. The service bureau can then control competing demands to use the gateway without being concerned that control points in other networks are setting up transforms in the gateway node for internetwork sessions. To accomplish this, one of the gateway SSCPs within a gateway can be designated solely responsible for controlling the gateway node.

When two networks are interconnected by more than one gateway, as illustrated in Figure 20, the resulting configuration is that of parallel gateways. Each gateway may be any one of the types previously described. The gateway nodes in each gateway can be controlled by the same gateway SSCP, which makes that SSCP a part of both gateways. Alternatively, each gateway node can be controlled by distinct gateway SSCPs. These alternatives are shown in Figure 21.

Multiple gateway nodes between networks allow alternative paths for both SSCP-to-SSCP sessions and

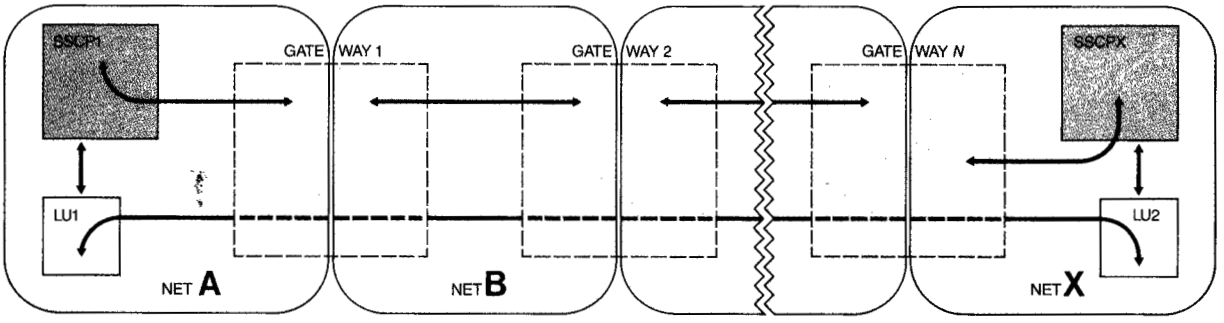
Figure 21 Alternative parallel gateway configurations



LU-to-LU sessions. When setting up a session, the gateway SSCP automatically tries each successive gateway node in a user-defined list until the session is active or until it determines that the session cannot be established. Thus, even if a gateway node is not operative or lacks alias addresses, the session can be activated via an alternate gateway node.

Networks attached to separate gateways can be interconnected, and any number of successive gate-

Figure 22 Cascaded gateways



ways can exist between them. Figure 22 shows cascaded gateways that allow a logical unit in NETA to have a session with a logical unit in NETX, even though there are several intermediate networks between NETA and NETX. Session setup and takedown requests are rerouted through the series of gateways on successive sessions between gateway SSCPs. Each of the gateways in the cascade can be any one of the types described above, and it is controlled according to the rules for its type. The network identifier and network name of the destination logical unit must be determined within the first gateway. There can be parallel gateways between any pair of the networks, and user-defined lists can cause a gateway SSCP to try alternative setup paths, just as in a two-network configuration.

Cascaded gateways allow one network to provide data transmission services between two other networks, without constraining all three to attach to a common gateway node. One network might even be used to transmit data between two parts of another network, as shown in Figure 23. Although it is possible that none of the routes in NETA is available for a session between LU1 and LU2, NETB may provide the necessary path for the session.

Two networks can interconnect and retain maximum isolation from each other by using a special case of cascaded gateways. Two gateway nodes, one in each network, are connected with SDLC links and are defined to create an intermediate network that consists only of the address space in the gateway nodes. Referring to Figure 24, gateway SSCPA in NETA controls gateway node 1, and gateway SSCPB in NETB controls gateway node 2. The intermediate network consists only of the address space in the subareas of the NETX portions of gateway nodes 1 and 2. Transforms are established in both gateway nodes for a session between logical units in NETA and NETB. Since neither network has to add a new subarea when interconnecting this way, there are fewer additions to routing tables at other subareas of the networks than with the other gateway configurations.

Design case study

During the requirements phase of the network-interconnection design, several networks were studied, to better understand the functions that were needed. Since IBM internal networks were already using an application pass-through technique for interconnection, they were used as case studies for requirements identification, design objectives verification, and design walkthroughs. This section summarizes the study of one specific network.

In 1981, that network supported more than 20 000 terminals attached by over 200 subarea nodes. Access was provided to more than 100 host processors and to a range of ancillary network services, such as multiplexed bulk data and message switching.

When the study was conducted, that network had been using an application pass-through technique for interconnection called Concentration/370 (CON/370). CON/370 handles the routing of messages through one or more network nodes using multiple Binary Synchronous Communication (BSC) links.[20] It also provides an interface that allows an SNA application to use its routing services to another SNA application or to a terminal.

At the time of the case study, the network consisted of thirteen geographically separate networks that were interconnected via a fourteenth central network. The size of the networks varied from one host

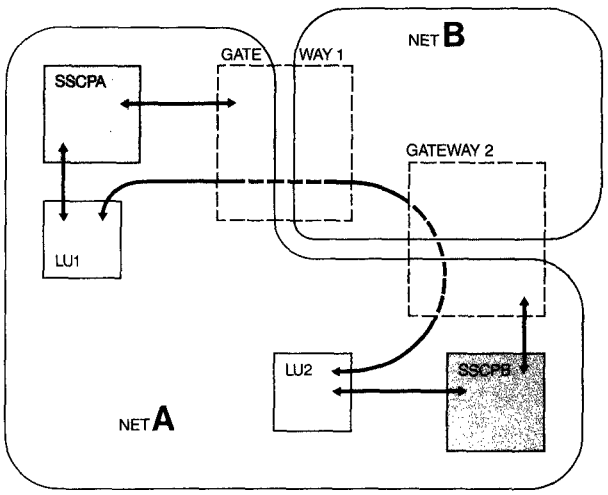
with one communications controller to twelve hosts with twenty communications controllers. Each of the networks was created and maintained individually, although conformance with certain organizational design standards was required.

The different sizes of the networks caused the address splits incorporated in these networks to vary, depending on the mix of terminals and subarea nodes. For instance, two of the networks each used a 6/10 network address split, allowing up to sixty-three subarea nodes, with a potential of 1024 elements per subarea. Most of the other networks used a 7/9 split.

In the light of their experience, the managers of the interconnected networks were asked to consider two alternatives: (1) using the SNA network interconnection techniques, or (2) combining all the networks into a single SNA network. They concluded that SNA network interconnection was the more appropriate solution. Their analysis, including the rationale for migrating from the current CON/370 technique, was as follows:

- They could remove the maintenance support for the CON/370 application.
- They could replace the Binary Synchronous Communication connection used for CON/370 with the more efficient Synchronous Data Link Control[21] for their internetwork communications.
- They would obtain such benefits from SNA as link sharing for applications, flow control mechanisms, and enhanced performance.
- They could interconnect with other IBM networks without difficult protocol negotiations.
- Amalgamation of the independent networks would be possible without changing any of their addressing structures.

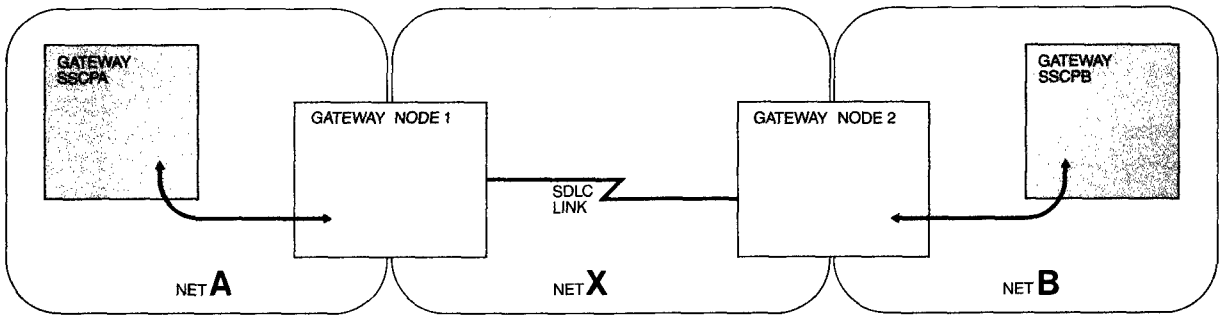
Figure 23 Cascaded gateways to connect disjoint parts of one network



- Local management could retain control of each network.
- Each network could independently expand its addressability and yet have adequate addressability to access the central network's applications and other networks' applications.

The original structure of the network made it easy to plan the migration and conversion to SNA network interconnection. The original network was organized with CON/370 nodes acting as Binary Synchronous Communication (BSC) gateways between the central network and other networks. The migration plan replaces the CON/370 nodes with appropriate SNA gateway nodes, which are connected using SDLC internetwork links instead of BSC

Figure 24 Cascaded gateways with minimal intermediate network



links. Added to this replacement topology are additional SDLC connections from some of the networks to multiple gateway nodes, one at each central site.

Because the gateway placement was rather simple to determine, the more difficult task was to plan a smooth and quick installation of the SNA gateway. The networks' management recognized that the evolutionary approach selected for gateway and back-level product support allowed them to gradually prepare each network for gateway usage prior to installing the gateway. They devised a migration plan that allowed them to install and activate the gateway rapidly without major disruption to the existing networks.

A migration plan allowed them to install a gateway rapidly without major disruption to the existing network.

According to the migration plan, prior to the installation of the gateway, management would rebuild the routing structure (path routing tables) within each network to reflect the addition of the SNA gateway node.

Management would then order and install the internetwork links between the gateway nodes in the central network and communications controllers in the other networks. These internetwork links—including appropriate IBM 3705 link scanners—would be generated into the network control programs, but not activated until the gateway nodes are installed. Given this plan for the topology changes to include the gateway nodes, the actual tasks are to regenerate the ACF/NCP nodes to include the new routing tables and hardware scanners, and to update the ACF/VTAM routing tables. This essentially ends the preparation for the physical part of the network, with the exception of installing the gateway nodes. Some planning for the logical network is still to be done.

After all the gradual system generations and reloading of the network nodes have been accomplished, the central sites can generate, install, and load the gateway hosts and gateway nodes without disrupting the operations of the other networks. All that will be needed are operational procedures to activate the internetwork definitions and the internetwork links. Any future additions to a network, or inclusion of a new network, will not necessitate the above changes. Only the gateway nodes and the individual network are affected.

The network studied has no need for the name translation function, because management intends to use unique network names throughout their interconnected networks. This is made possible by assigning network names using a naming convention. Although the name translation function will not be used in the network studied, management intends to install it to support connection with a corporate-wide network. This will eliminate any problems relating to the duplication and usage of network names for internetwork sessions.

Concluding remarks

The SNA network interconnection function is designed to be introduced with minimum disruption to existing networks, yet it offers the advantage of access to a much enlarged population of terminals and application programs. Autonomy of network operations and internal network protocols is preserved. By using the gateway as a building block, SNA networks can be interconnected in a variety of configurations to suit individual network needs.

Acknowledgments

The collective authorship of this paper is but a small indication of the collective effort that created what the paper describes. Aiding in the acquisition of the requirements, and at times advising on the technology, were Guy Benson and Ed Michels from the Communication System Programming Requirements area, Phil Grise from the Marketing Division, and Larry Korn from the Corporate Consolidated Network. Aiding in the creation and selection of the technology were Tom Taylor and Robert P. Lee from Communication System Program Design; Jim Romano from NCCF; Barb Heldke from VTAM; Fred George from TCAM; Roy Hayward from NCP; Ellis Miller, Joe Rusnak, Fred McGriff, Jim Gray, and John Torrey from Communications System Architecture; and Frank Moss, Mike Conner, and

Parvis Kermani from the Research Division. Aiding in the design validation, as well as providing design guidance, were Bob Wagner from the Corporate Consolidated Network and Giovanni Nasti from Information Systems Europe. Many others, not named here, refined the high-level design and created the actual products described by this paper.

Cited references and notes

1. R. J. Sundstrom and G. D. Schultz, "SNA's first six years: 1974–1980," *Proceedings of the Fifth International Conference on Computer Communication*, Atlanta, GA, October 27–30, 1980, pp. 578–585.
2. *Systems Network Architecture Concepts and Products*, GC30-3072; available through IBM branch offices.
3. J. P. Gray and T. B. McNeill, "SNA multiple-system networking," *IBM Systems Journal* **18,** No. 2, 263–297 (1979).
4. Recommendations X.75 and X.121 of CCITT, 1980 Plenary of the International Telephone and Telegraph Consultative Committee.
5. C. Sunshine, "Current trends in computer network interconnection," *Eurocomp 78: Proceedings of the European Computer Congress*, London, England, May 9–12, 1978, On-line Conference, Ltd., Uxbridge, England (1978), pp. 465–472.
6. J. B. Postel, "Interconnected protocol approaches," *IEEE Transactions on Communications* **COM-28,** No. 4, 604–611 (April 1980).
7. Current host nodes that implement subarea routing include the IBM System/370, the IBM 4331, 4341, 303X, and 3081 running ACF/VTAM, ACF/VTAME, ACF/TCAM, or ACP access methods. Current communication controller nodes include the IBM 3705 and 3725.
8. V. Ahuja, "Routing and flow control in Systems Network Architecture," *IBM Systems Journal* **18,** No. 2, 298–314 (1979).
9. CICS is an SNA application program. For further information see 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).
10. M. Gien and H. Zimmerman, "Design principles for network interconnection," *Sixth Data Communications Symposium*, November 27–29, 1979; IEEE Cat. No. 79CH1405-0, pp. 109–119. This paper includes "level of interconnection" in its discussion of design principles.
11. Logical Unit Level: An end user, rather than communicating directly with an end user in another network, communicates with an intermediate application in the gateway. This application pass-through technique has been used for several years.
12. Explicit Route Level: In this alternative, only the explicit route identifiers (origin subarea, destination subarea, and explicit route number) are changed from network to network. This gives a virtual route endpoint the illusion that the virtual route is local to the network of that endpoint.
13. The gateway SSCP is implemented in ACF/VTAM Version 2, Release 2. The name translation functions of the gateway SSCP are implemented in NCCF Version 2.
14. The gateway node is implemented in ACF/NCP Version 3.
15. For information about the program products that comprise the gateway, see *Network Program Products General Information*, GC27-0657; available through IBM branch offices.
16. F. D. George and G. E. Young, "SNA flow control: Architecture and implementation," *IBM Systems Journal* **21,** No. 2, 179–210 (1982).
17. R. A. Weingarten, "An integrated approach to centralized communications network management," *IBM Systems Journal* **18,** No. 4, 484–507 (1979).
18. For information about the relationship between NLDM and other network program products, see Reference 15.
19. R. A. Weingarten and E. E. Iacobucci, "Logical problem determination for SNA networks," *IBM Systems Journal* **22,** No. 4, 387–403 (1983, this issue).
20. *General Information—Binary Synchronous Communications*, GA27-3004; available through IBM branch offices.
21. R. A. Donnan and J. R. Kersey, "Synchronous data link control: A perspective," *IBM Systems Journal* **13,** No. 2, 140–162 (1974).

Reprint Order No. G321-5199.

Jay H. Benjamin *IBM Information Systems and Technology Group, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. Benjamin joined IBM as a programmer in 1968. He worked on graphics application programs for the IBM 1130/2250 and on enhancements to the OS/360 Graphics Access Method. From 1972 until 1981, he participated in the design and development of VTAM and helped to design SNA multisystem network support, mesh-network support, and interconnected-network support. He now works in the field of systems structure and architecutre. Mr. Benjamin received a B.A. degree in mathematics from Harpur College of the State University of New York in 1967.

Matthew L. Hess *Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Dr. Hess joined IBM in 1968 as a systems engineer. He specialized in performance analysis and in the design of on-line systems. In 1976, he joined the telecommunications center in La Gaude, France. Later, he came to Research Triangle Park, North Carolina, where he analyzed various new public data network offerings. For the past four years, he has participated in the development of Systems Network Architecture. Dr. Hess received a B.Eng. from McGill University, Montreal, Canada, in 1964. He received an M.Sc. in 1966 and a Ph.D. in 1968 from the University of Birmingham, England.

Robert A. Weingarten *IBM Corporate Headquarters, Old Orchard Road, Armonk, New York 10504.* Mr. Weingarten joined IBM in 1969 in the former IBM New York Development Center, where his assignment was on the OS/360 linkage editor. He joined the U.S. Army in 1970. Upon returning to IBM in 1972, he worked on the DOS RPG II compiler. In 1974, he transferred to Kingston, New York, where he was involved in various aspects of the definition of Systems Network Architecture, including high-level systems design, systems requirements gathering and planning, and system design management for the Advanced Communication Function access methods and communication network management. In 1982, he was the control program design and development manager in the scientific and engineering processor development area. Since March 1983, he has been assigned as a consultant on the Engineering, Programming, and Technology corporate staff, concentrating on communication programs. Mr. Weingarten received his B.S. and M.S. degrees in electrical engineering from New York University in 1967 and 1969, respectively.

Walter R. Wheeler *Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Wheeler joined IBM in 1966. His first assignments were those of programming for the Basic Telecommunications Access Method (BTAM) and for the Queued Telecommunications Access Method (QTAM). Since 1970, Mr. Wheeler has done programming and design for the Network Control Program (NCP) of the IBM 3705 Communications Multiplexor. He has worked on all releases of NCP, specializing in implementation of Systems Network Architecture by NCP products. Mr. Wheeler was principally involved in the design of SNA support for dial lines and for the dynamic reconfiguration function. He is currently working in the field of communication architecture. Mr. Wheeler received his B.S. degree in mathematics education from the State University College at New Paltz, New York. He is actively involved in the Special Olympics program in North Carolina.