SNA Management Services architecture for APPN networks

by M. O. Allen
S. L. Benedict

The introduction of Advanced Peer-to-Peer Networking (APPN) provides for a more flexible Systems Network Architecture (SNA) environment: end-user systems (physical units, or PUs) at the edge of a routing network no longer need a predefined relationship with a system services control point (SSCP) for network control purposes. This new flexibility creates challenges for SNA Management Services, however, since the SSCP-PU relationship provided a vehicle for network management as well as network control. To meet the needs of this peer-to-peer environment, the SNA Management Services architecture was extended to provide a management infrastructure that replaces the previous SSCP-PU relationship, and at the same time provides for much greater flexibility. This new infrastructure consists of a formalization of the focal-point/entry-point concept in the architecture and a transport technique for management services data that utilizes the facilities of Advanced Program-to-Program Communications (APPC) rather than the SSCP-PU session. Together this provides for a management structure in a peer network.

In 1991, IBM extended Systems Network Architecture (SNA) by the announcement of Advanced Peer-to-Peer Networking (APPN), which provides distributed networking with dynamic topology awareness, automated resource directory, dynamic route calculation, and network congestion control. These advanced networking functions simplify system definition, increase reliability, and provide network flexibility for APPN resources.

The development of APPN necessitated a new infrastructure for SNA Management Services (SNA/MS) to replace the relationship that previously existed between the system services control point (SSCP) and physical unit (PU). The new infrastructure consists of the focal-point/entry-point relationship and a transport mechanism that uses the facilities of Advanced Program-to-Program Communications (APPC). This infrastructure is much more dynamic, flexible, and robust than the previous SSCP-PU relationship. Significant efforts have been made to ensure that the new infrastructure is economical in its use of network resources. The new structure has been designed for growth. New management functions and application programs can be added easily to existing ones, without modification to the infrastructure.

This paper describes enhancements to the SNA/MS architecture that have been implemented by a number of IBM products. One of those product implementations, NetView*, is described in a paper by Irlbeck[1] in this issue.

The previous hierarchical SNA environment

Before the introduction of APPN, SNA/MS used the relationship that exists between the peripheral nodes of an SNA network (type 2.0 and boundary-function attached type 2.1 nodes) and their "owning" SSCP. The SSCP "activates" a peripheral node by transmitting a request unit (RU) named Activate Physical Unit (ACTPU) to the node. If the command is accepted by the peripheral node (signified by a positive response to the ACTPU), the SSCP-PU session becomes active. After activating the PU, the SSCP then activates each logical unit (LU) at the peripheral node by transmitting the Activate Logical Unit (ACTLU) command. The SSCP-LU session is used for session control purposes only, not by management services.[2]

The SSCP-PU session is the mechanism for sending commands or requests for management data from an operator to the PU, and for returning responses and data from the PU to the operator. The SSCP is also the target for unsolicited management data created by the PU, such as alerts.[3] Management services data are transported over the SSCP-PU session in the SNA request unit called the Network Management Vector Transport (NMVT). The actual management data are encoded using management services major vectors, which follow the NMVT header in the RU.[4,5]

This close tie between management services and the SSCP-PU relationship simplified SNA/MS, since the management services configuration and data transport session were already provided. However, over time, this close relationship became restrictive. The SSCP-PU session placed severe limitations on the amount of data that could be transmitted by management application programs. As the quantity and diversity of management data continued to grow, this limitation became a significant problem for programmers who were developing new management application programs. Additionally, the session is limited with regard to the correlation of requests (such as an operator command) and the subsequent replies. Only one request may be outstanding at a time over a given session, thus forcing all management services application programs at a PU to effectively run in a serial fashion.

A more important restriction was the fact that management services were available only to those systems which were part of the SSCP configuration. Figure 1 illustrates how it became possible to have a network containing SNA nodes that could not be effectively managed.[6] The SNA nodes on the local area network (LAN) that are "downstream" from the boundary-function attached T2.1 node are not recognized by the SSCP and therefore they cannot be effectively managed. Limited management of these resources may be accomplished by a variety of "pass-through" techniques, but generally these mechanisms are very limited in their capabilities.

While the restrictions of the SSCP-PU relationship presented problems for management services, these restrictions were not significant enough to justify the development of an entirely new architecture for an SNA/MS infrastructure. Each of these limitations could be (and has been) circumvented in specific environments in a number of different ways.[7] However, the introduction of APPN required a completely new management infrastructure, since the SSCP-PU relationship does not exist in peer networks.

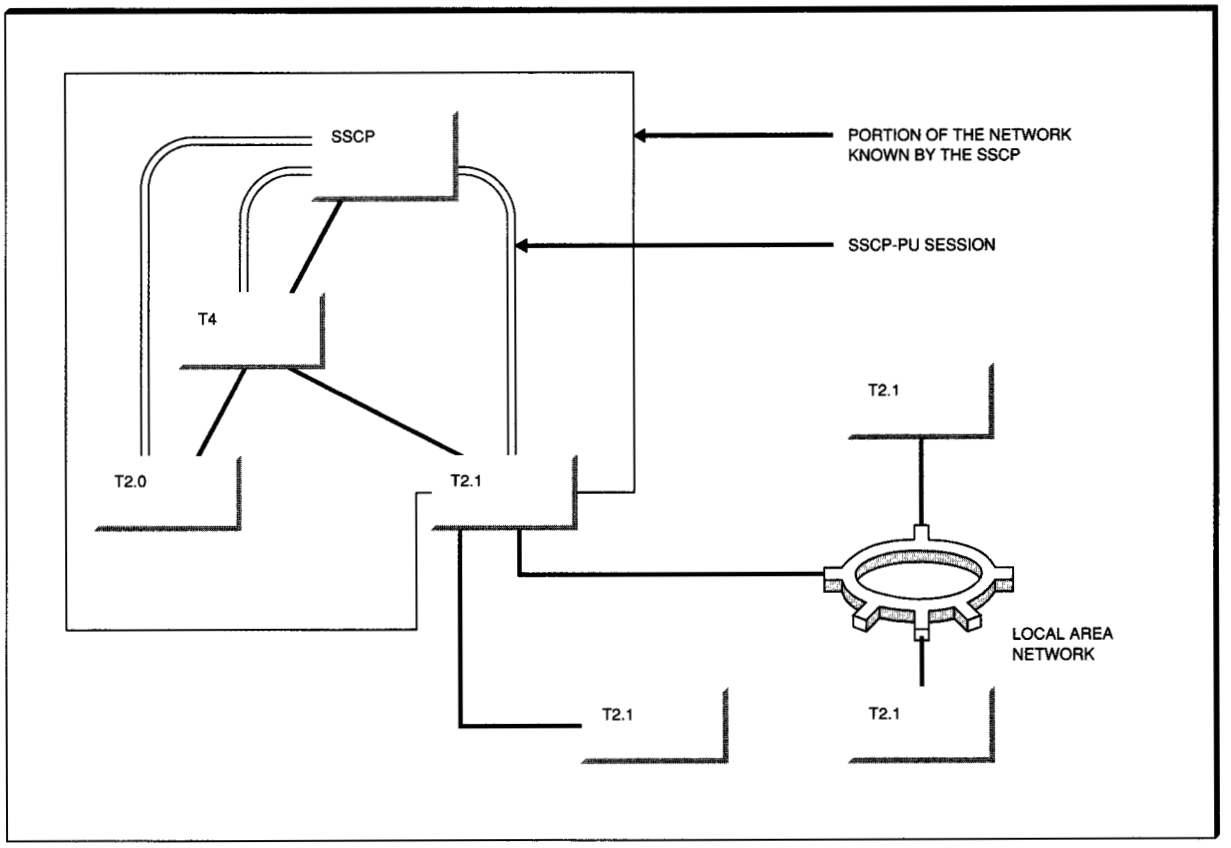
The new peer-to-peer SNA environment

The APPN architecture for T2.1 nodes[8] defines three types of peer nodes that can participate in Advanced Peer-to-Peer Networking:

- Low-entry networking (LEN) node—The basic T2.1 node peer protocols required for participation as an end point in APPN
- APPN end node—An enhanced LEN T2.1 node that provides additional function (for example, sessions between T2.1 control points) that allows it to participate, more fully, in APPN
- APPN network node—An enhanced T2.1 node that provides all APPN functions, including network services (for example, route calculation) to attached APPN end nodes and LEN nodes

APPN networks may consist of any number of LEN nodes, APPN end nodes, and APPN network nodes. LEN and APPN end nodes participate only as session destinations or origins, and rely upon adjacent APPN network nodes to provide the required networking services. APPN and LEN nodes are connected by links using a variety of protocols, including token ring, X.25,[9] IBM System/370* channel, and SDLC (synchronous data link control).

Figure 1 Configuration restrictions on the SSCP-PU session



Every APPN node contains a control point (CP) that participates in the following control services with other nodes in the network, through CP-CP sessions using LU 6.2 protocols:

- Connection—Connecting a new link or node into an existing network
- Directory—Building and maintaining a directory of local and remote network resources (logical units) and participating in the distributed APPN directory activities for locating session partners
- Route selection—Maintaining the network topology database among the interconnected network nodes and determining the preferred route through the network to meet a requested class-of-service
- Session—Initiating, negotiating, and activating sessions between two logical units
- Data transport—Controlling data traffic flow in the network, based upon session priorities, and providing congestion control through adaptive pacing
- Management—Performing network problem management and sending alerts to a network focal point when the node detects programming or machine problems

Figure 2 shows the relationship of network nodes to end nodes in an APPN network. While APPN provides connectivity for LEN nodes, these nodes do not participate in CP-CP session, and SNA/MS support for them is limited.

APPN ease-of-use and reliability are achieved by: (1) distributing network control so that each APPN node can participate in peer-to-peer network operations, (2) automating the directory look-up, route calculation, session binding, and congestion control processes, making them transparent to APPN network users, and (3) using network control and data transport algorithms that adapt

traffic flow, automatically, to accommodate dynamic APPN network conditions.

Challenges for management services in APPN

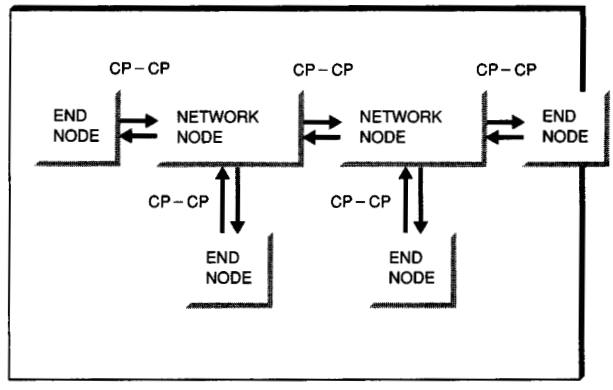
APPN provides for true peer-to-peer networking with completely distributed control: there is no centralization of network control, as there was previously with the SSCP. While this new method of peer networking has significant advantages for building and maintaining networks, it forced the development of a new SNA/MS infrastructure. The requirements for this infrastructure were to maximize configuration flexibility, minimize network overhead, coexist in existing SNA networks, and allow future growth.

Maximize configuration flexibility. Users indicated that they needed flexibility in their management configurations. For some enterprises, completely centralized management was ideal, while other enterprises preferred to distribute most network management responsibilities to regional or organizational centers. Another requested option was to provide flexibility to organize management by categories, so that problems could be reported to one system, while other categories of management data from the same nodes, such as performance statistics or accounting data, could be collected by different systems or at different locations. Users wanted the option to alter their management configurations nondisruptively based on the time of day or the day of the week, or to bypass systems during planned outages. Finally, when failures occur, the management configuration must recover itself automatically.

Minimize network overhead. The use of LU 6.2 sessions[10] for MS data transport to replace the SSCP-PU was a logical choice for several reasons:

1. LU 6.2 sessions are not restricted by configuration in the same way that SSCP-PU sessions are. They are supported in a wide variety of SNA networks, such as subarea networks, LEN networks, and APPN networks.
2. The data-encoding method for LU 6.2 transaction programs, the generalized data stream variable (GDS), permits records of up to 32 767 bytes to be sent and received very simply. The LU 6.2 architecture provides the facilities that segment outbound application program data streams and then perform the appropriate reassembly for inbound data. Data records longer than 32 767 bytes are sent as a series of 32 767 records.[11]
3. LU 6.2 sessions may be nonpersistent, meaning that they may be deactivated when no longer needed, and reactivated again when needed, thus minimizing costs in networks using public switched telecommunications facilities. This deactivation and reactivation of sessions is accomplished transparently to application programs using LU 6.2 services.
4. LU 6.2 includes provisions for session-level security (password authentication of session partners), an important consideration in more open peer-to-peer networks.[12]

Figure 2 CP-CP sessions are used for network control in an APPN network

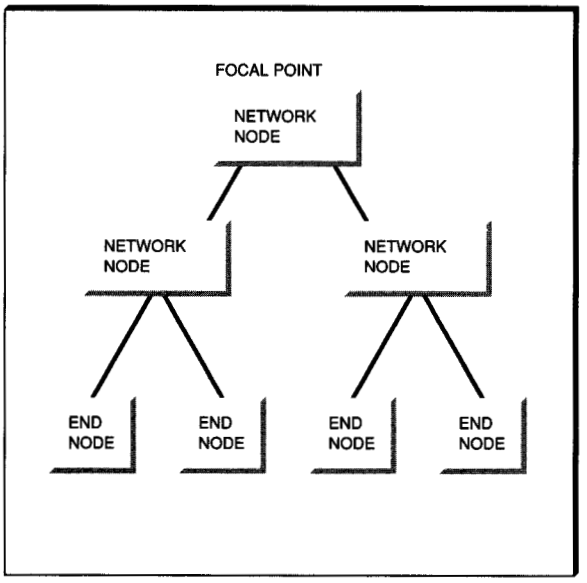


However, early studies indicated that careless use of LU 6.2 sessions by management services might impose unacceptable levels of overhead in very large networks, especially during critical times such as startup. The management services architecture had to find ways to minimize the overhead imposed on a network by the transport of management data.

Coexist with management in existing SNA networks. Since many users have significant investments in existing SNA networks, both subarea and APPN, any new management services infrastructure must accommodate and coexist with these networks.

Enable future growth of management application programs. We recognized that SNA/MS (and the products that implement the architecture) must

Figure 3 Single focal point in an APPN network



provide a platform on which existing MS functions can easily coexist with new management application programs, whether defined by SNA/MS, individual IBM products, other vendors, or IBM customers.

Focal-point architecture

The concept of a focal point was developed by SNA/MS architecture to allow for centralized management of a distributed or peer network, such as an APPN network. A focal point is an application or set of applications that provides centralized management and control for other applications (entry points) for one or more network management categories. A category is an architected management services function, for example *alert* or *operations management*. Applications for each category come in pairs, one for the focal point, and one for the entry point. The focal-point architecture allows for a management structure in a peer networking environment.

For example, in an APPN network made up of network nodes and end nodes, one of those nodes could be designated as the focal point for alerts. As the nodes communicate with the network, a focal-point/entry-point relationship is established with the focal point, and it is known which of the many peer nodes in the network will be serving as their alert manager. This communication mechanism is detailed in the section on MS Capabilities exchanges. A simple network with a single focal point is depicted in Figure 3. The focal point would have a focal-point/entry-point relationship with all the nodes it is responsible for managing. The nodes may be either network nodes or end nodes. The network node and its served end nodes can be treated as a single unit from the perspective of the focal point.

A communications network may have multiple focal points. This configuration may be used for distributed management, for session concentration, or business reasons. These focal points may have responsibility for the same or different management services categories. For a particular category of data (e.g., alerts) focal points may be peers, that is, all at the same level. For example, alerts may be distributed among various focal points based on type of equipment, business unit, etc. Figure 4 illustrates a multiple focal-point network. Each focal point would be responsible for managing a portion of the network. Alternately, they may be related hierarchically. When a focal point itself has a focal point (other than itself), it is in a configuration where it is the nested (or lower-level) focal point and its focal point is the nesting (or higher-level) focal point. This may be to distribute the management closer to the source, sending only particular types of problems on to the nesting focal point. Or it may be to distribute automation and forward only when human intervention is required. Figure 5 illustrates a nested focal-point network. The responsibility for network management could be divided among the four nested focal points. The two nesting focal points could manage the nested focal points directly, and the rest of the network indirectly.

MS Capabilities exchanges. An entry point that forwards information, especially unsolicited information, needs to know where to send the information. Nested focal points have the same need. In a peer networking environment it is important to establish where the management will take place. Specific policy will determine the placement of the focal-point applications and the knowledge of the location of the focal point. MS Capabilities exchanges were developed to provide the mechanism for establishing these relationships.

Figure 4 Multiple peer focal points in an APPN network

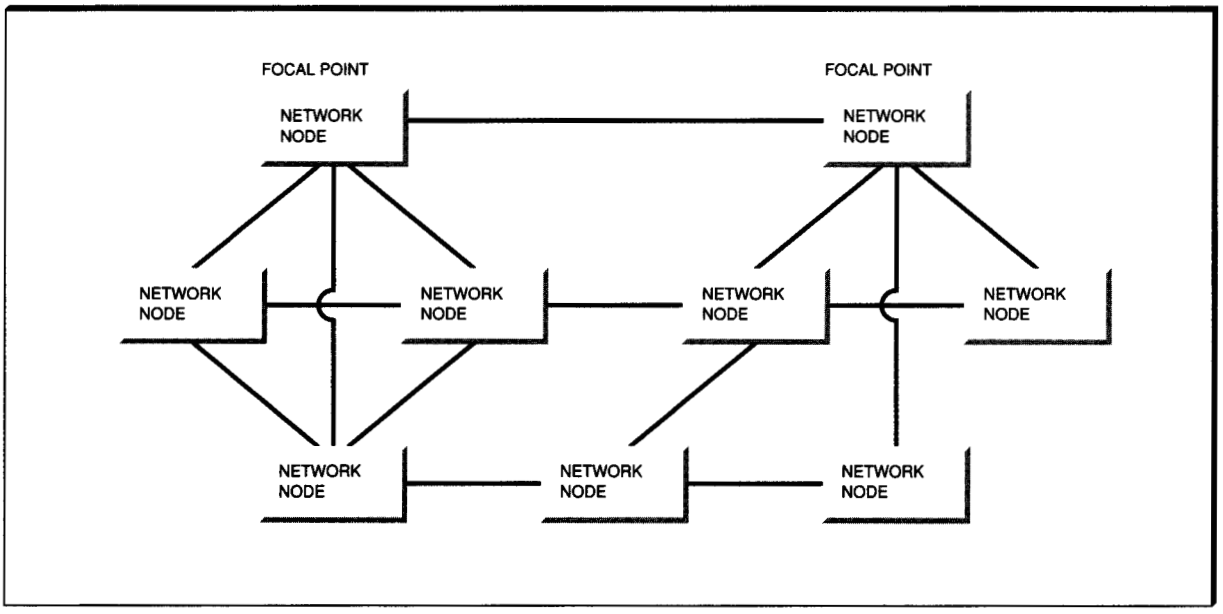
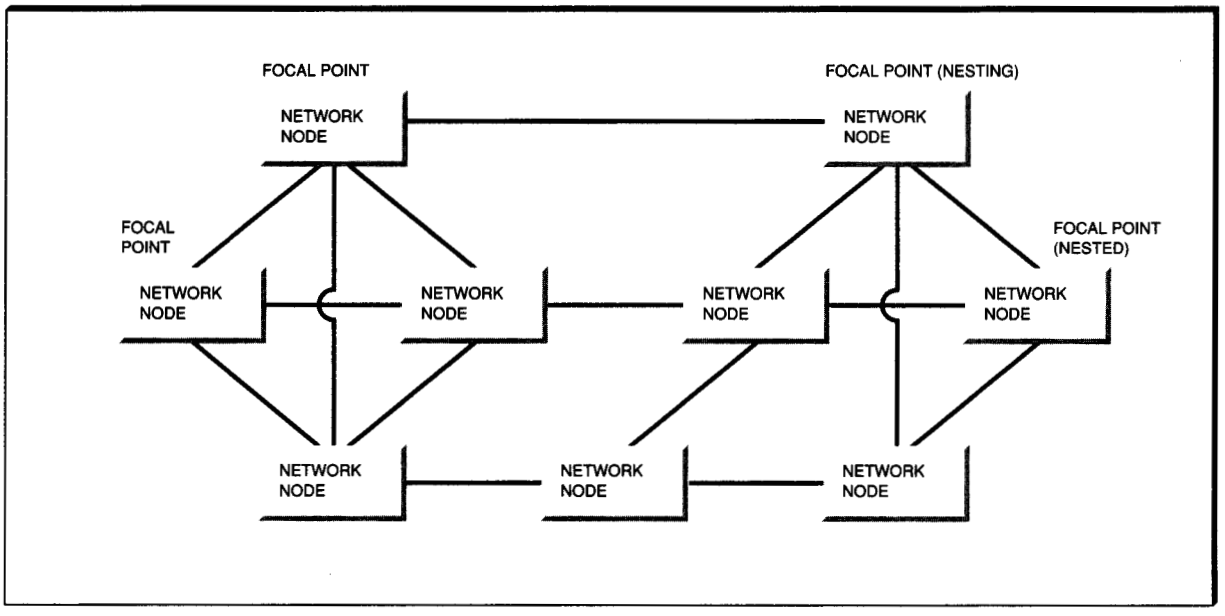
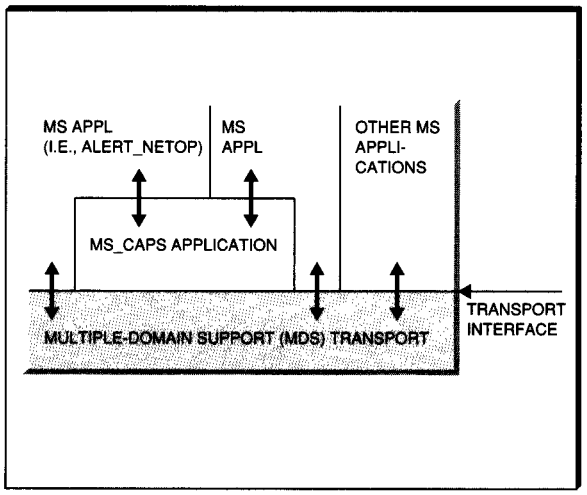


Figure 5 Nested focal points in an APPN network



Either the focal point or the entry point can initiate the sending of MS Capabilities, and therefore can be responsible for establishing the focal-point/entry-point relationship. Either point may also be responsible for recovery of the relationship, should communications fail between the two. Both are discerned from the MS Capabilities message. The determination of which node will initiate and re-

Figure 6 MS_CAPS application concept



cover is made by the policy mentioned earlier. Either the entry point knows the identity of the focal point, or the focal point knows the identity of the entry point. In either case, each searches for the other to establish this relationship as soon as possible after it is activated using MS Capabilities. An entry point may support having backup focal points, so that in the event of a failure in communications with the primary focal point, it can locate the backup and request focal-point services from it. This is also accomplished via MS Capabilities exchanges, as are the flows to recover the services of the primary focal point.

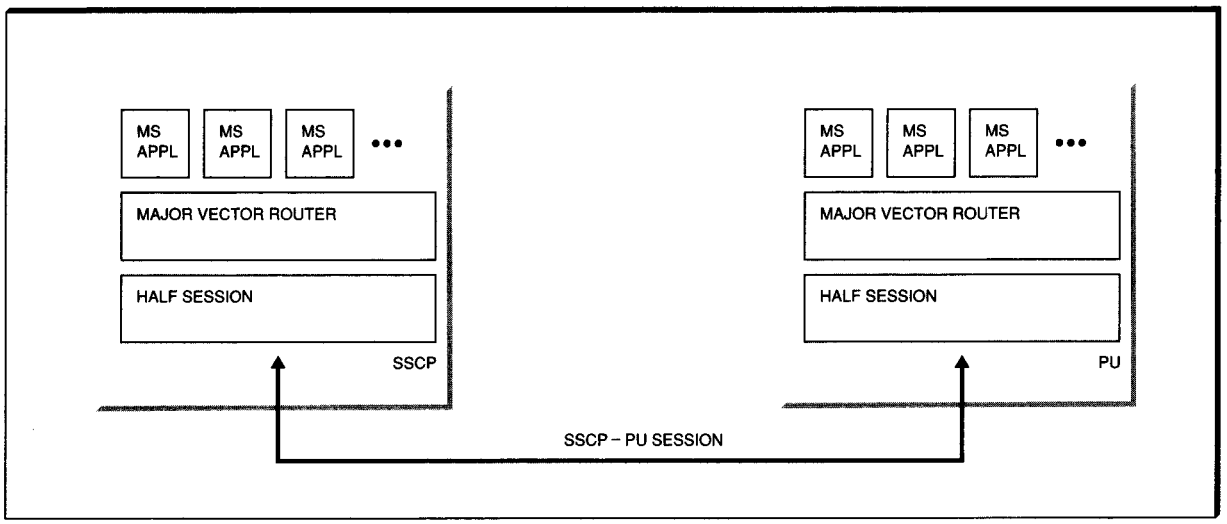
MS_CAPS application. A node may have more than one application that is interested in focal-point information for a given category. Additionally, there may be more than one application category interested in maintaining focal-point identification information on any given node. It could be a costly proposition if every application were required by itself to participate in the message exchange for obtaining this focal-point information, to maintain this information, and to use it to dynamically keep a current focal point. Therefore, a single application, called MS_CAPS, provides these services on behalf of all interested applications, whether they be focal-point or entry-point applications.

The MS_CAPS application has a transaction program name that is registered in the architecture. Every APPN node is expected to have one of these MS_CAPS applications present as part of its management services functions. There are different levels of function that exist for this application, depending on the type of node it resides on (e.g., a network node or an end node). Also there are optional functions that can be implemented on either base. This "base and towers" structure provides for implementation flexibility, while ensuring a consistent base.

MS_CAPS centralizes the processing of MS Capabilities messages, which establishes a management structure in a peer network. It handles the establishment of these relationships dynamically. MS_CAPS also provides a common service for maintaining information on primary and backup focal points. This facilitates the sending of unsolicited data and forwarding of data in an SNA network. Upon detection of an error condition, it dynamically changes from primary to backup and back again. This can occur without any impact to the applications using the service, and may even occur before they know there is a problem. This also provides for a centralized place in a node where any application can get focal-point information. Applications that want to use this service need only register with MS_CAPS. The architecture describes protocol boundaries for applications to register with MS_CAPS. They can register as a focal-point application, as an entry-point application (that is, one that wants to receive focal-point information for a certain category) or both. The MS_CAPS application concept is depicted in Figure 6. The MS applications (MS Appls) register with MS_CAPS. MS_CAPS and the applications use the MDS (multiple-domain support) transport service for sending and receiving data.

Conceptually, the same process occurs automatically for end nodes from their serving network node as if the end node registered as an entry point for all categories. However, they do not have to be registered. The end node will forward all focal-point information and store it in a local MS Capabilities table. Interested applications must be registered at the end node with their MS_CAPS application to receive the focal-point information. The mechanism for forwarding this focal-point information from serving network node to end node is via a specific type of MS Capabilities message.

Figure 7 Relationship of the SSCP-PU session to MS application programs



An MS Capabilities table is maintained by the MS_CAPS component. All focal-point identification information for this node for all MS categories and any user-defined categories is maintained there. Information that is kept is the category name, the rank (which is used in determining whether to accept or reject another focal-point authorization request), the focal-point name (network name, control-point name, and application name), the backup focal-point names, if applicable, the list of applications that have registered to be notified about focal points, and the list of served end nodes (if this is a network node) or the serving network node (if this is an end node).

Nested focal-point processing. As mentioned previously, a focal point can itself have a focal point. When this occurs, it continues to serve as the focal point for the nodes it was managing. However, the focal-point application on this node will now be directly managed by the nesting (higher-level) focal point. This relationship allows the lower-level (nested) focal point to forward information to its higher-level (nesting) focal point. Nodes being managed by the nested focal point do not have any direct knowledge of the nesting focal point. They only have visibility to their first-level focal point. Likewise, the nesting focal point does not have direct knowledge of the nodes that the nested focal point is managing. The nested focal point does have knowledge that it is a focal point, and continues to act as focal point for itself, its local applications, and any nodes it is currently serving.

Transport architecture

The SSCP-PU session. The previous method for transporting SNA/Management Services records between the host NetView product and other systems is the SSCP-PU session. This single session is shared between various management services application programs and other SNA control functions. The relationship between the management services application programs and the session is illustrated in Figure 7. The data records for a number of MS application programs are passed on a single session. Records are routed to the appropriate application program by record key (MS major vector key).

While the sharing of the SSCP-PU session in this manner does minimize the number of sessions that might otherwise be required, the nature of the session imposes restrictions on the management services application programs that use it. The session restricts the length of records transmitted over it to 256 bytes in the outbound direction (SSCP to PU) and to 512 bytes in the inbound direction (PU to SSCP). Automatic segmentation and reassembly of basic information units (BIUs) to and from smaller path information units (PIUs) is not provided on this session as it is on the LU-LU

sessions typically used for transmitting application program data. The SSCP-PU session also has limited correlation capability for application programs, so only one request may be outstanding on a session at a given time. These restrictions impose significant design constraints for management services application programs.

Use of LU 6.2 sessions. Enhancing the characteristics of the SSCP-PU session was not a long-term solution to these problems, since these sessions are not necessarily available in APPN networks. Instead, SNA/Management Services was enhanced to exploit the general-purpose characteristics of LU 6.2 for transporting MS records. Transaction programs utilizing LU 6.2 are not subjected to the kinds of restrictions associated with the SSCP-PU session, such as severe limits on record sizes. The services of LU 6.2 deal with these restrictions on behalf of the application program (referred to, in SNA context, as a transaction program, or TP, in the LU 6.2 architecture) for example, by dividing a large application program record into smaller segments for transmission over a session. The receiving side reassembles these segments back into the original record before presenting it to the receiving transaction program. This segmentation and reassembly is performed transparently to the transaction programs. Another difference between the SSCP-PU session and LU 6.2 sessions is that the SSCP-PU session is full duplex, meaning that records can be sent and received at the same time, while LU 6.2 sessions are half duplex, meaning that records cannot be sent and received at the same time. However, this limitation of LU 6.2 is easily overcome, since LU 6.2 supports the use of parallel sessions between transaction programs, effectively permitting the full-duplex exchange of records between programs.

Multiple-domain support. A number of alternatives using LU 6.2 were evaluated. The transport technique that was ultimately developed contains the following elements:

- Eliminates current SSCP-PU message size restrictions on management services application programs
- Achieves full-duplex throughput between management services application programs in two nodes
- Minimizes the number of sessions required to support management services in a network
- Allows for multiple outstanding request/replies between two nodes and simplifies the correlation of application program data

Figure 8 illustrates the basic concepts of the solution, which was named *multiple-domain support* (MDS).

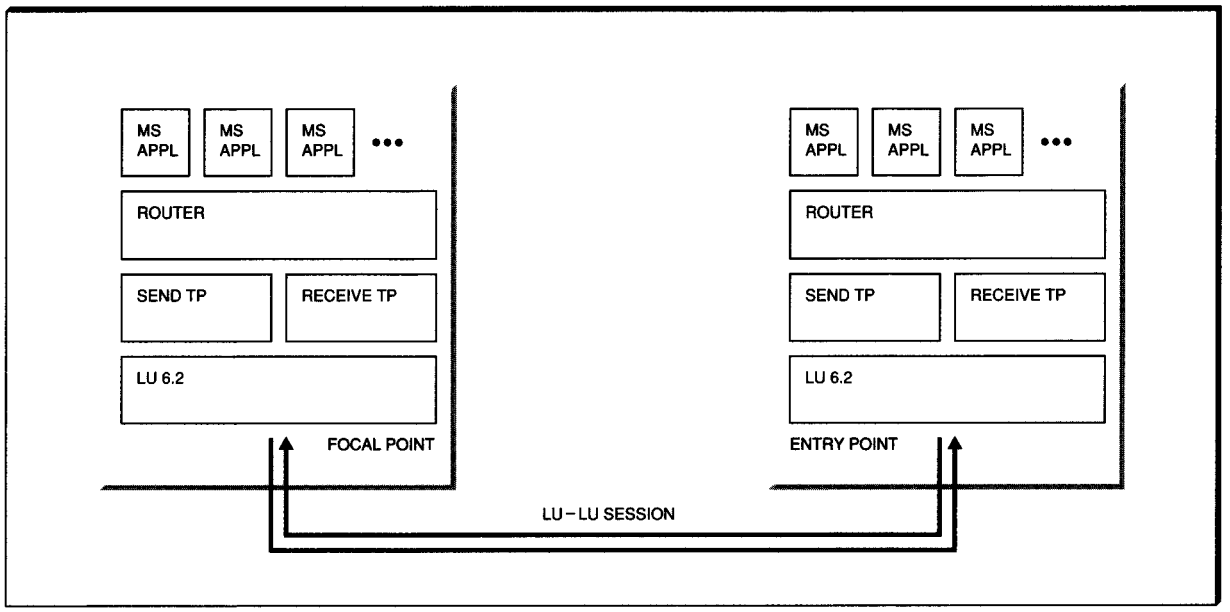
In Figure 8, *focal point* indicates a node with management services application programs that provides centralized operations for other nodes, the *entry points*. These terms are more generic than SSCP, which may be a focal point, and PU, which is an entry point.

The basic concept of MDS is the interleaving of management services application program records from a number of application programs onto shared LU 6.2 conversations.[13] The application programs are not LU 6.2 transaction programs (i.e., they do not issue LU 6.2 verbs); instead, they interact with a common MDS router service to send and receive their records. A single instance of a utility transaction program sends the records for one or more application programs over a single LU 6.2 conversation. At the target node, a single instance of a utility transaction program receives the records and passes them to the MDS router, which performs the appropriate intranode routing. The interleaving of application program records on conversations in this manner eliminates the requirement for multiple parallel sessions to support simultaneous application program conversations.

By using a pair of LU 6.2 sessions, the send utility transaction program in each node can be active at the same time, sending records simultaneously to the other node. The receive transaction program in each node will also be active at the same time, simultaneously receiving records from the other node. This capability to simultaneously send and receive allows MDS to provide full-duplex throughput between two nodes.

The architecture uses the MDS-MU encoding described in a following section on encoding of management data for MDS. The fields in the MDS header identify the origin and destination nodes and application programs for the message and provide the correlation value used to associate replies and error messages with corresponding requests.

Figure 8 Basic concept of multiple-domain support



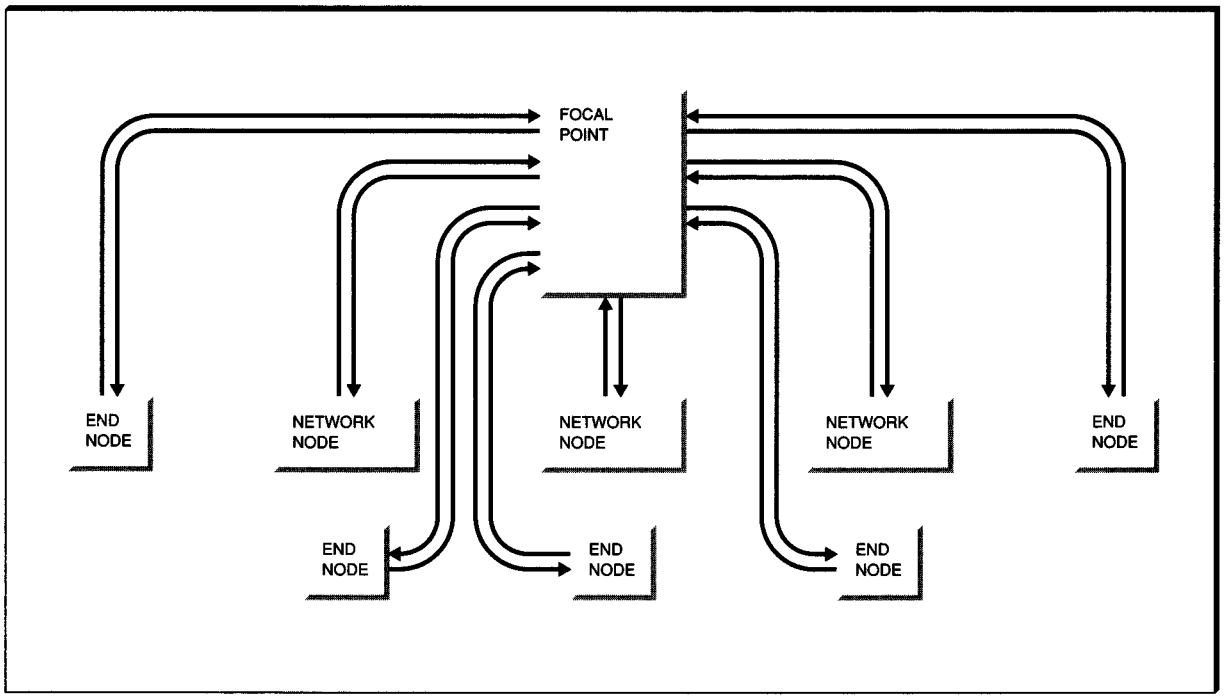
The base architecture for MDS provides a high level of transport reliability through the use of session-level confirmations, guaranteeing that MDS-MUs will not be lost during transport. If an MDS-MU cannot be delivered to the indicated destination, an MDS error message is sent back to the origin indicating the nature of the problem.

The base architecture for MDS strives to minimize the overall use of network resources for the transport of management data. It does this by sharing the same LU 6.2 sessions that are used for network control. Since these sessions are shared, MDS minimizes the length of its conversations over these sessions, so that other SNA transaction programs (such as those for APPN control functions) can use the same sessions. This design is optimized for the most economical use of resources, and the analysis of management application program requirements indicates that this is the most appropriate optimization for most applications.

However, some applications were identified that were best served by a different optimization. These were application programs that needed to transmit a very large number of records quickly with minimal delays. The conservation of network resources was secondary in these applications to the need for maximum effective throughput. In order to meet these requirements, an MDS optional subset was developed, describing a variant of MDS that products may implement if needed to provide for very high throughput application programs. (In SNA/MS, an optional subset is the documentation method for describing sets of functions from which implementing products may choose, based upon functions that they need; as contrasted with a base subset, which is required function.) The trade-offs made to achieve this higher level of throughput were the elimination of session-level confirmations for MDS-MUs (the application program must implement its own mechanism, such as sequence number checking, to verify that records are not lost), and the use of dedicated LU-LU sessions.

Optimization of MDS for APPN. A study of SNA networking trends leads to the conclusion that future APPN networks will be very large. In those networks, the ratio of APPN end nodes to network nodes will be high, i.e., each network node will serve a large number of end nodes. It is also expected that the end nodes will produce relatively small volumes of management services records. In that environment, the number of LU 6.2 sessions required to support direct connectivity between focal-point nodes and APPN end nodes for MDS would be comparatively high, and the utili-

Figure 9 LU-LU sessions that would have been required without MDS optimization for APPN networks



zation of those sessions will be comparatively low. This situation is undesirable for the end nodes, which should be relieved of as much SNA control overhead as possible, and is undesirable for the focal-point nodes, which must activate and maintain very large numbers of relatively unproductive sessions.

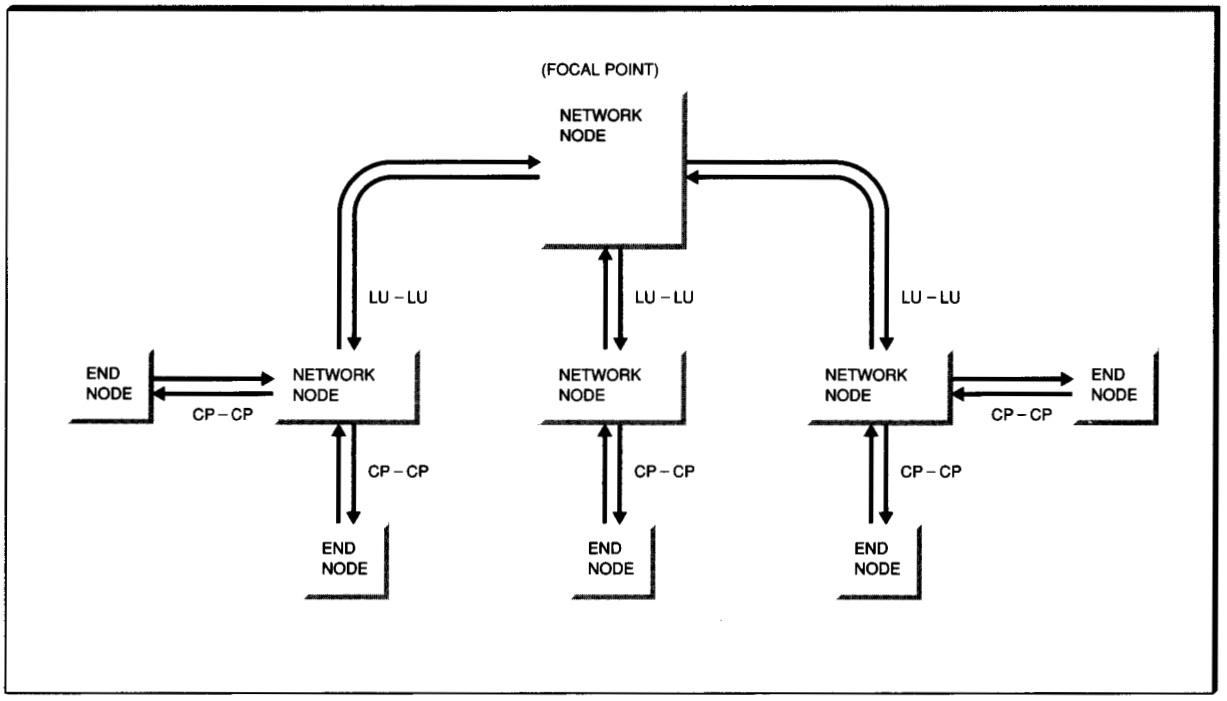
In APPN networks, LU 6.2 is already utilized for the control-point-to-control-point sessions (CP-CP) sessions[14] between end nodes and their serving network nodes. MDS as described so far does not take advantage of these existing sessions, since a focal point would communicate with all other nodes (including APPN end nodes) via LU-LU sessions. The number of control sessions supported by each end node would thus be increased, an impact that is inconsistent with the APPN goal of minimizing end node overhead.

An APPN network is illustrated in Figure 2. In that figure, the arrows labeled CP-CP represent the APPN control sessions between nodes. Figure 9 illustrates the MDS sessions that would be required if a new node, acting as a focal point, were introduced to the previously illustrated APPN network. Although they are not shown in Figure 9, the CP-CP control sessions are still required by APPN as well.

MDS was optimized for APPN in the following ways:

- Management services data are sent between an APPN end node and its serving network node over existing CP-CP sessions. Separate sessions for management services are not required.
- APPN network nodes provide routing of management services data for both their local MS applications program and for management services application programs on served end nodes.
- Focal-point nodes send management services data destined for APPN end nodes over LU-LU sessions to the appropriate serving network nodes, which then forward that data to the end node over the CP-CP session.
- All management services routing decisions are based upon dynamic information available from the APPN directory of network resources. No statically defined routing information is required.

Figure 10 Concentration of MDS sessions in an APPN network



A single pair of LU-LU sessions from a serving network node to a focal-point node can be used to transport all management services data between the focal point and all the end nodes served by the network node, as well as that between the focal point and the network node itself.

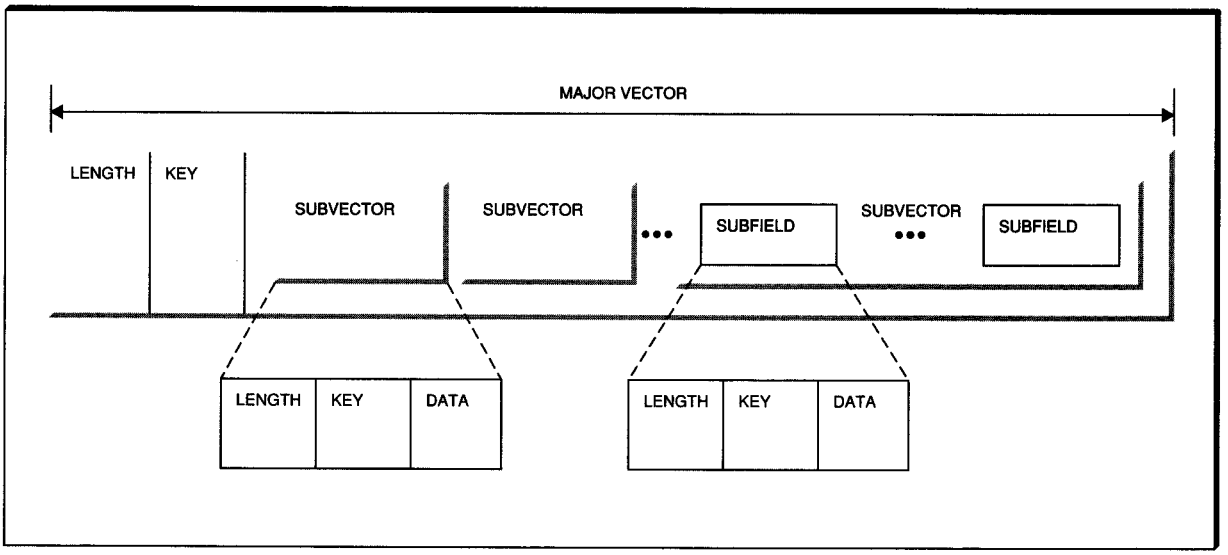
The reduction in the number of sessions required is illustrated in Figure 10. This figure shows the same network in Figure 9, except that now the focal-point node only has to support LU-LU sessions to the network nodes, not to all the end nodes. In this trivial example, the number of session pairs required at the focal point was reduced from eight to three. The savings are more dramatic with a more realistic example: for a modest network of 20 network nodes with 20 end nodes supported by each network node, the number of session pairs supported by a single focal point would be reduced from 420 pairs to just 20 pairs.

In addition to the savings in the number of sessions supported by a focal point, each end node is saved from having to support another pair of SNA control sessions. However, direct LU-LU communication with a focal point may sometimes be required because of configuration constraints. So end nodes must be able to support the LU-LU sessions as well, when the sessions are initiated by the focal point.[15]

Encoding of management data for MDS. Previously, when an SNA/Management Services request or data record was received by a node, the internal routing of the record was performed by examining the record type. This technique was adequate for the simpler SNA networks of the past, in which the only management services functions were those provided by the networking products themselves (in general, only limited application programming interfaces were provided). However, the networking products of today are providing more open application interfaces to support network management functions developed by other parties. The use of record types to perform internal routing of management services records is inadequate in this environment.

The previous method for selecting the appropriate processing for an SNA/Management Services record is based upon the record type. Figure 11 illustrates the general scheme for encoding data in

Figure 11 Overview of a management services major vector/subvector/subfield structure



SNA/MS. The major vector key field identifies the record type.

The limitations of routing records by major vector key value are:

1. Migration and testing of management services functions can be difficult, since multiple copies of a particular function cannot easily be executed in the same system.
2. In some cases, the installation of particular software products together in the same system must be restricted because the products all need to operate upon the same MS record types.
3. Management services functions are restricted to data encoded in the major vector format, since the major vector key is needed for routing. If MS functions need to exchange data that are not in major vector format, new major vector keys must be defined in each case just to envelope the data.
4. Unique sets of major vectors are required for each management services function. Since major vector keys are used for internal routing, an existing major vector cannot be reused in a new context. New major vectors must be defined for each new function, even if an existing format would be satisfactory.

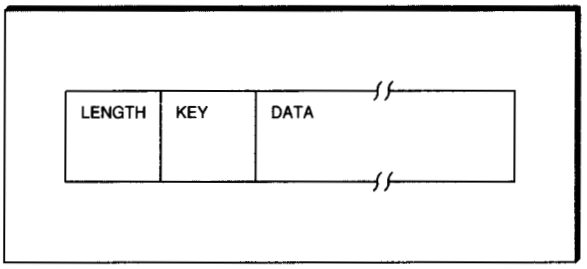
The limitations of the major vector routing technique are overcome by the use of explicit application program names for routing to MS functions within a system. MS application program names follow the LU 6.2 TP naming technique, which provides for four-byte architecturally-defined values or one- to eight-character installation-defined names (LU 6.2 base support).[12]

The MDS router provides a service for sending and receiving records for the MS application programs. Each application program registers its name with the router so that incoming records will be routed correctly.

For each MS request or data record, the origin and destination MS application program names are identified in fields of a common message header. All routing of messages to MS functions is performed by examining the fields in the common header. The actual MS request is enclosed within a generalized data stream (GDS) variable that follows the header (see Figure 12). Since the contents of the MS request or data are no longer examined for internal routing purposes, MS functions have the freedom to reuse existing major vector formats for new purposes, or, if appropriate, use data encoded using techniques other than major vectors. Refer to Figure 13 for an overview of the encoding used to accomplish the routing of MS messages, the Multiple-Domain Support Message Unit (MDS-MU).

The MDS-MU contains the MS multiple-domain support header and MS application program data. The MS multiple-domain support header consists of (1) MDS Routing Information—the information used by multiple-domain support for routing between control points and between MS application programs—and (2) an Agent Unit of Work Correlator—used to correlate requests, replies, and error messages. MS application program data consist of a GDS variable supplied by the MS application. The variable may be a CP-MSU, as shown in Figure 14, an SNA Condition Report, or some other GDS variable.

Figure 12 Structure of a generalized data stream (GDS) variable



The maximum size permitted for the application program logical record in the MDS-MU is 31 743 bytes. This restriction is imposed to simplify implementation design, so that the entire MDS-MU (MDS header and application data) will always be no more than 32 767 bytes in length.

MDS message types. MDS supports the following message types (message type is indicated by settings in the Flags subvector of the MDS header): MDS request, MDS reply, and MDS error message.

An MDS request initiates a transaction. It may optionally be followed by one or more MDS replies. The router maintains awareness of outstanding request/reply transactions so that the appropriate error reporting to the application programs (via the MDS error message) can be performed if a failure impacts the successful completion of the transaction. The request/reply correlation mechanism provided by MDS is more robust and flexible than the NMVT correlation which takes place on the SSCP-PU sessions. MDS allows any number of transactions to take place concurrently between two nodes. On the SSCP-PU session, only one transaction can be active at a time.

The MDS error message. The MDS error message is the vehicle for reporting errors detected by the MDS component at any point along a transaction's path. It is also used by application programs in a few cases. Figure 15 shows a high-level decomposition of an MDS error message. Note that the

Figure 13 Structure of the Multiple-Domain Support Message Unit (MDS-MU)

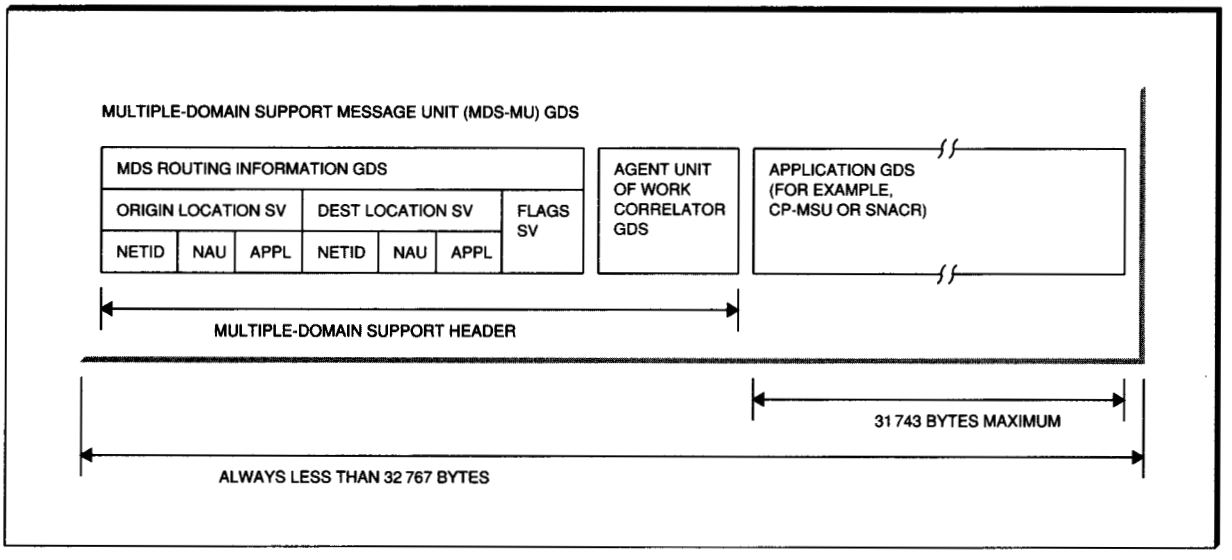


Figure 14 Structure of a Control Point Management Services Unit (CP-MSU)

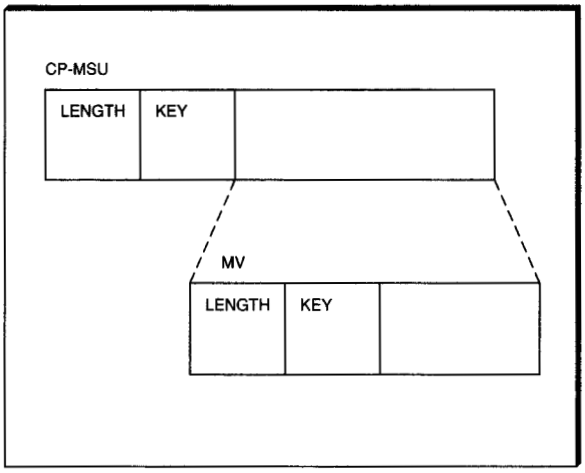
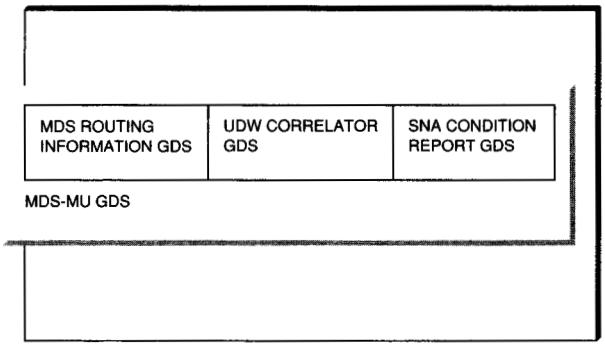


Figure 15 The MDS error message



application program GDS is an SNA Condition Report rather than a CP-MSU.

MDS error messages are created by MDS for two categories of errors: (1) routing errors, in which an individual MDS Message Unit could not be delivered successfully to its destination application, and (2) transaction failures, where the error pertains not to an individual MDS Message Unit, but to the sequence of such message units that comprise a transaction. For example, it is a transaction failure if the MDS router is able to determine, from either the receipt of session outage notification or the expiration of a timer, that no reply is forthcoming for a request that expected one.

The MDS error message has the following characteristics:

- It is sent either from the MDS router in the node that detected the error, or from one of the communicating MS application programs.
- The Agent Unit of Work in the MDS header is the one for the transaction that failed.
- The application program information is an SNA Condition Report, which carries an SNA report code (i.e., sense data) identifying the precise error that was detected, plus the NAU name and application program name for the other application that was involved in the transaction.

Application program-level error reporting. The MDS error message is typically not used for reporting application program-detected errors. These errors, i.e., command reject, parsing exception, and function not supported, are reporting via application-defined techniques.

However, there are circumstances under which an MS application program must be able to unconditionally terminate an outstanding MDS transaction. For example, an MS application program may start a timer when an MDS request is sent to another MS application. If no reply has been received when the timeout period has elapsed, the sending MS application program may conclude that something is wrong with the destination application, causing it not to respond. Since the sending MS application program is not going to wait for the reply any longer, it terminates the outstanding transaction with an MDS error message, allowing both the partner application program, as well as MDS, to understand that transaction is not active.

An MDS error message is also used to indicate that an unsupported application program GDS has been received. These messages would not be able to use the usual "function not supported" application error reporting technique, since the format of the request itself is unsupported.

Conclusion

The introduction of APPN, with its distributed control, necessitated the creation of a new infrastructure for SNA Management Services to replace the hierarchical SSCP-PU relationship. The new SNA/MS infrastructure consists of the focal-point/entry-point relationship, formalized in the

MS Capabilities architecture, and a transport mechanism, using LU 6.2 capabilities, called multiple-domain support. This infrastructure is much more dynamic, flexible, and stronger than the previous SSCP-PU relationship. Significant efforts were made to ensure that the new infrastructure was economical in its use of network resources. Finally, the infrastructure was designed for growth: new management functions and application programs can be added easily alongside existing ones, without modification to the infrastructure.

Future direction for APPN management

As announced in March of 1992, IBM's general direction is to provide applications that use Open Systems Interconnection (OSI) management standards to provide APPN management. Protocols based upon OSI Common Management Information Protocol (CMIP) transported over LU 6.2 sessions (exploiting MDS, as described previously in this paper) will be used for communications between managing and managed system applications.

The initial set of applications using OSI management will provide for the reporting of dynamic APPN network node and end node topology to a manager system, and will provide the collection of accounting information for APPC (LU 6.2) resources to a managing system.

Acknowledgments

The authors wish to express their appreciation to all their coworkers who played such an important part in the development of this architecture: Bob Moore, Steve Golberg, Mark McKelvey, John Wilder, Tony Amitrano, Fred Tsai, Robert Nielsen, Leon Proulx, Bob Clouston, Jeannette Lee, Larry Plank, and Curtis Frantz. Thanks also to Ray Boyles for helping us on this paper by contributing the APPN overview material.

Cited references and notes

1. B. W. Irlbeck, "Network and System Automation and Remote System Operation" *IBM Systems Journal* **31**, No. 2, 206–222 (1992, this issue).
2. *SNA Technical Overview*, GC30-3073, IBM Corporation; available through IBM branch offices.
3. The one exception to the exclusive use of the SSCP-PU session in SNA/MS is change management, which uses SNA Distribution Services, which in turn uses LU-LU sessions to transport data. See Reference 4 for additional information.
4. *SNA Management Services Reference*, SC30-3346, IBM Corporation; available through IBM branch offices.
5. R. E. Moore, "Utilizing the SNA Alert in the Management of Multivendor Networks," *IBM Systems Journal* **27**, No. 1, 15–31 (1988).
6. The capability of type 2.1 nodes to attach directly to one another using peer-to-peer protocols is referred to as low-entry networking (LEN). For more information, consult Reference 2.
7. An example of a circumvention of the SSCP-PU restrictions within SNA/MS can be found in the Common Operations Services function set, which includes a mechanism for the segmentation of certain major vectors across a series of NMVTs, thus bypassing the session restriction on message length. See Reference 4 for details.
8. *SNA Type 2.1 Node Reference*, SC30-3422, IBM Corporation; available through IBM branch offices. For an overview of APPN, see also Reference 2.
9. X.25 is the ISO Standard for interface to packet-switched communication system.
10. The terms *LU 6.2* and *Advanced Program-to-Program Communications* (APPC) are often used interchangeably. However, strictly speaking, LU 6.2 refers just to the architecture, while APPC refers to both the LU 6.2 architecture and its various implementations in products. This paper generally refers to the LU 6.2 architecture.
11. These limits on GDS variable length apply to LU 6.2 transaction programs using basic conversations, such as SNA service transaction programs. User-written application programs typically use mapped conversations, which do not impose these limits.
12. *SNA LU 6.2 Reference: Peer Protocols*, SC31-6808, IBM Corporation; available through IBM branch offices.
13. A *conversation* is a logical connection between two transaction programs using an LU 6.2 session. For more information, see Reference 12.
14. See Reference 2 for more information about CP-CP sessions.
15. In a mixed subarea/APPN network, MDS at a subarea focal-point node is unable (since it does not have access to the APPN resource directory) to determine the appropriate routing of an MDS-MU to an APPN end node. However, a direct LU-LU session is possible between the focal point and the end node, so the communication can take place. This limitation applies only to transactions initiated from the focal point. When the transaction initiates at the end node, the MDS-MUs are routed correctly through the network node to the focal point.

Accepted for publication February 3, 1992.

Michael O. Allen *IBM Networking Systems, 200 Silicon Drive, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Allen joined IBM in 1980 with the IBM Information Network in Tampa, Florida. While in Tampa, he worked in the development of customized networking software, in particular an early mechanism for the interconnection of SNA networks called TRAP. (This project was described in a paper by K. D. Ryder, "An Experimental Address Space Isolation Technique for SNA Networks," *IBM Systems Jour-*

nal **22**, No. 4, 367–386, 1983.) Mr. Allen came to Research Triangle Park, North Carolina, in 1984, joining the Network and Systems Management Architecture and Standards organization, where he has worked on a number of projects associated with SNA management services architecture. Since 1987 he has specialized in architecture for the management of APPN networks, and was responsible for the development of the multiple-domain support architecture described in this paper. He is currently involved in the application of OSI management standards to the management of SNA networks.

Sandra L. Benedict *IBM Networking Systems, 3039 Cornwallis Road, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mrs. Benedict joined IBM in 1985 with SNA Architecture and Telecommunications. She worked in the Network and Systems Management Architecture organization where she was responsible for external publication of the SNA Management Services architecture. In 1987 she assumed responsibility for the architecture for the management of APPN networks, specializing in focal-point architecture. Since 1989 she has been in the Network Management Products organization, where she has been involved with the design and development of APPN network management in the NetView program product over a number of releases. She is currently the lead designer for APPN network management for the NetView product area.

Reprint Order No. G321-5477.