An experimental address space isolation technique for SNA networks

by K. D. Ryder

The integration of computer networks has led to increasingly large and complex configurations. This integration has resulted in concern for the availability of resources for the merged networks. An experimental technique called TRAP has been developed as a way to minimize the constraints on such networks. It allows address space isolation between interconnected networks. This paper discusses the technique and its fundamental process of network address translation.

The IBM Systems Network Architecture (SNA) provides a comprehensive conceptual base for communications networks. The IBM Advanced Communications Function (ACF) represents a product implementation of those concepts. Through SNA and ACF, single-host and multiple-host networks can be merged into increasingly functional and complex topologies. As larger and more diverse networks are merged, certain aspects of such integrated network environments must be carefully considered. This paper defines some significant considerations related to merging networks and describes an experimental technique called TRAP that has been developed in response to those considerations.

A major benefit of SNA is the enhanced sharing of network resources. This sharing provides better utilization of such resources and also improves communication opportunities. These same considerations motivate the integration of SNA networks. Network mergers maximize both the benefits of resource sharing and the potential for communications.

Although the benefits of network mergers are real, problems may be posed by network integration.

- When networks are merged, certain resources must be shared among those networks. One such resource is the pool of network addresses. Since this resource is finite, merged networks will tend to deplete the supply of network addresses. Furthermore, each network is constrained by the common usage standards for network addresses.
- Another shared resource is the pool of network names. Each name that is to be known throughout an integrated network environment must be unique. Although the potential name space is typically much larger than the potential address space, network name management is still an important consideration. The set of possible names is finite, and name usage must be carefully controlled to eliminate ambiguities.

Thus, the tangible benefits of network integration lead to increasingly tangible difficulties as that integration becomes more extensive.

The merger, or integration, of SNA networks typically incorporates those networks into a single entity. In the past, these mergers have been achieved through use of the ACF Multisystem Networking Facility (MSNF). In some multiple-network environments, it is desirable to communicate between networks, but it is not desirable to completely merge

networks. The recently announced SNA Network Interconnection function represents an IBM strategic product response to such requirements.[1]

TRAP is an earlier, simpler approach to the interconnected network environment. TRAP allows SNA networks to communicate while still retaining some degree of autonomy within each network. For the IBM Information Network, TRAP has provided an interim capability which will ultimately be migrated to the full SNA Network Interconnection function.

Although TRAP provides significant address space isolation between interconnected networks, it does not provide name space isolation.

TRAP overview

The IBM Information Network has a significant interest in network interconnection. This function is required to support both the internal operations and external offerings of the Information Network.

In response to this interest, an experimental technique has been devised by personnel of the Information Network to minimize the constraints on network connectivity. This approach allows substantial address space isolation between interconnected SNA networks. At the same time, full resource sharing and communication between those networks is retained. These effects are achieved by providing address transformations at the boundaries between interconnected networks. The algorithm of the IBM Information Network for executing these transformations is called TRAP, since cross-network communications are *TRAPped* and translated as they transit network boundaries.

Following are the specific objectives for TRAP:

- To conserve the addressing capability of each interconnected network by concealing much of the address space of other networks
- To ease the interconnection of networks by allowing each network to retain its own address space and addressing conventions

Although TRAP provides significant address space isolation between interconnected networks, it does not provide name space isolation. The usage of network names must be coordinated among all the interconnected networks.

TRAP is implemented via modifications to the ACF Network Control Program (ACF/NCP), which is software that operates in an IBM 3705 Communications Controller.

General design considerations. The TRAP mechanism provides a solution to the current needs of the IBM Information Network. It does not attempt to provide a generalized capability applicable to other environments. Some of the most significant TRAP design considerations and constraints are now described.

Since TRAP is not a general-purpose solution, it cannot be dependent on modifications or extensions to SNA. Accordingly, TRAP does not change the transmission formats or protocols used between nodes in an SNA network.

For the same reasons noted above, no TRAP support is included in the base ACF products. All changes to those products have been made by the TRAP project as unique IBM Information Network modifications.

Maintainability considerations caused the TRAP design to focus on a single network component. All TRAP modifications are limited to ACF/NCP. Within the ACF/NCP itself, the TRAP design is based on a minimum number of hook points within the control program. Wherever possible, TRAP function is provided by discrete modules that operate outside the normal ACF/NCP flows.

Only limited resources were available for TRAP development. Accordingly, the TRAP design had to be implemented by a small project team. This consideration was addressed by focusing the design on only those basic functions most important to the IBM Information Network.

IBM announced a Network Services offering in September 1982. This offering allows user networks to attach to the IBM Information Network. User networks have access to Information Network com-

munications facilities and network management services. Also in September 1982, the Information Network was selected to provide a specialized service for the insurance industry. This service, the Insurance Communications Service, allows insurance companies and agents to communicate via the Information Network. As with the Network Services offering, the Insurance Communications Service allows the attachment of user networks.

TRAP is essentially transparent to data and control flows through the network.

It was essential that a viable address space isolation mechanism be available to support these new services. These critical schedule considerations served to focus the design and implementation on the specific capabilities required by these new services.

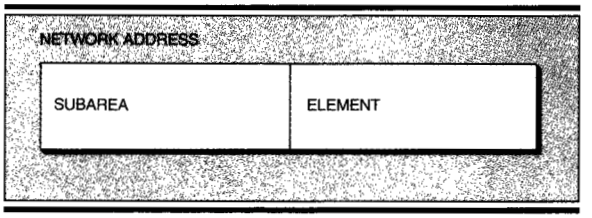
TRAP must not interfere with the normal communication between end users in an SNA network environment. TRAP is essentially transparent to data and control flows through the network.

The TRAP environment must be manageable. Therefore, the design preserves as many of the normal SNA network characteristics as possible. Network operators in an interconnected environment will perceive few differences because of the presence of TRAP.

Although TRAP is a special-purpose mechanism with limited functional objectives, it must provide those functions in a completely reliable manner. Accordingly, major emphasis was given to a formal design and review process and to the inclusion of diagnostic tools and aids.

Concepts and terminology. This section provides background material which is essential to a detailed discussion of TRAP. Some concepts and terminology are related to standard features of SNA and ACF. Other concepts and terminology are specific to the TRAP environment.

Figure 1 SNA address structure



It is not intended to provide rigorous, broadly applicable definitions. These concepts and terminology are introduced solely to facilitate a discussion of TRAP. A summary of key terminology may be found in the Appendix.

SNA address structure. TRAP is primarily concerned with address transformations. Therefore, a discussion of TRAP function requires some knowledge of the SNA address structure. A few general concepts, as opposed to formal architectural definitions, will provide an adequate context for this paper.

SNA addresses define the location of communications resources within a network. These resources normally include terminals, application programs, links, and other physical and logical entities.

A network address is composed of two parts—a subarea portion and an element portion as shown in Figure 1. The subarea identifies a major node within a network, such as a host or a communications controller. The subarea address is essential to routing message units through the major nodes of the network. The element portion of a network address identifies a specific resource, such as a terminal or application program. The element address is essential to routing message units within a major node.

The overall size of the network address is fixed. Only the dividing line between the subarea and element portions is selectable by each particular network. Once the network address subarea/element structure has been decided for a given network, that address split applies to all nodes within the network.

With standard MSNF integration of SNA networks, all network addresses must observe the same structure. In addition, network address usage must be

Figure 2 ACF multidomain environment

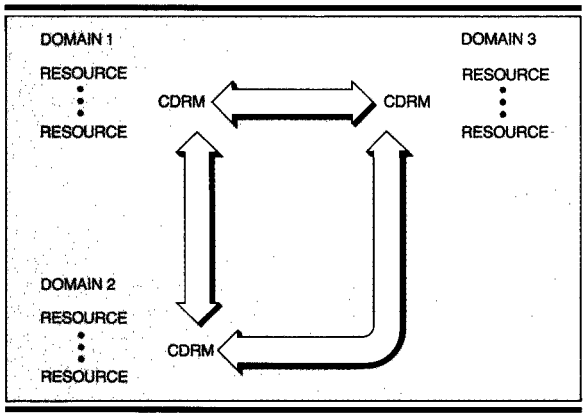
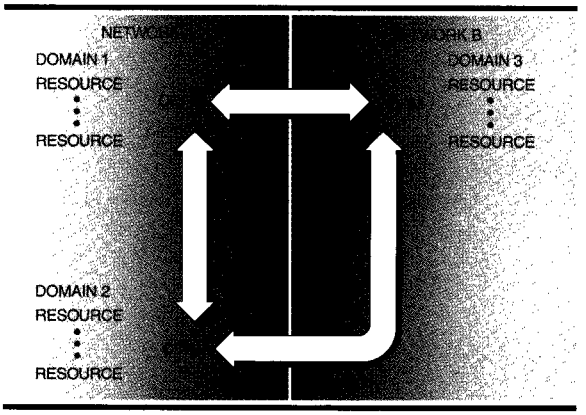


Figure 3 TRAP multinetwork/multidomain environment

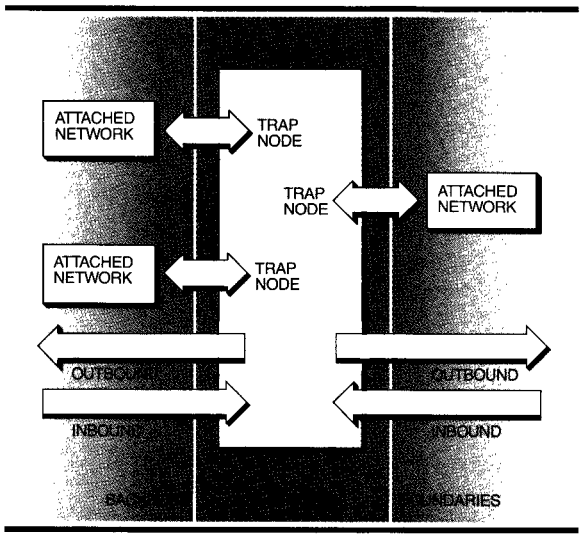


unique. The same network address cannot be simultaneously assigned to multiple resources.

TRAP creates an environment in which each network may independently select its own network address subarea/element structure. Each network may also independently assign addresses to its resources.

ACF multidomain environments. A discussion of TRAP also requires some familiarity with basic concepts of MSNF.

Figure 4 TRAP multinetwork environment (connectivity/flows)



An ACF *domain* is the set of logical and physical resources that is controlled by a single resource manager. A given network may contain multiple domains. Resource usage between domains is controlled via communication between pairs of *Cross-Domain Resource Managers* (CDRMs). In SNA, the logical connection between a resource pair is called a *session.* CDRMs, therefore, manage the establishment of *cross-domain sessions.*

CDRM function is typically resident in a System/370 CPU. CPUs in a network environment are referred to as *host* nodes.

Although MSNF allows multiple domains to be combined into a single network, each domain must adhere to a common set of network addressing standards. In addition, the resources of each domain must be allocated addresses from the common pool of network addresses. The common network address space limits the growth of existing domains and constrains the incorporation of additional domains. Figure 2 depicts this ACF environment.

TRAP multinetwork/multidomain environments. In the remainder of this paper, the term *network* refers specifically to a communications environment consisting of a single address space. If networks can communicate among themselves and still retain their individual address spaces, a multinetwork environment exists. Flows and functions that operate across a network boundary are *cross-network.*

TRAP allows substantial address space independence among interconnected networks. Each net-

Figure 5 Functions of TRAP Node

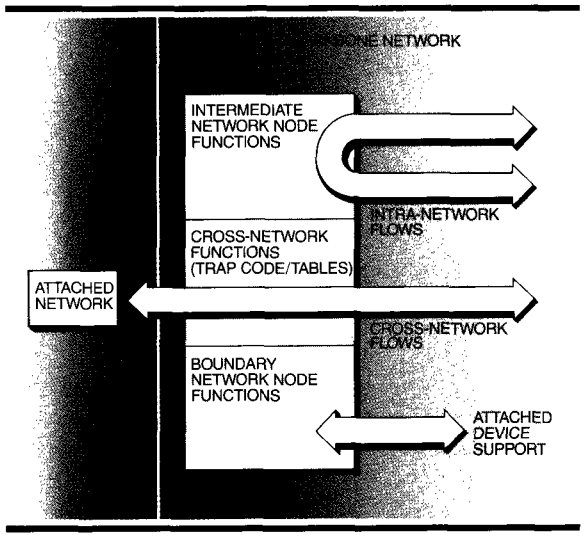
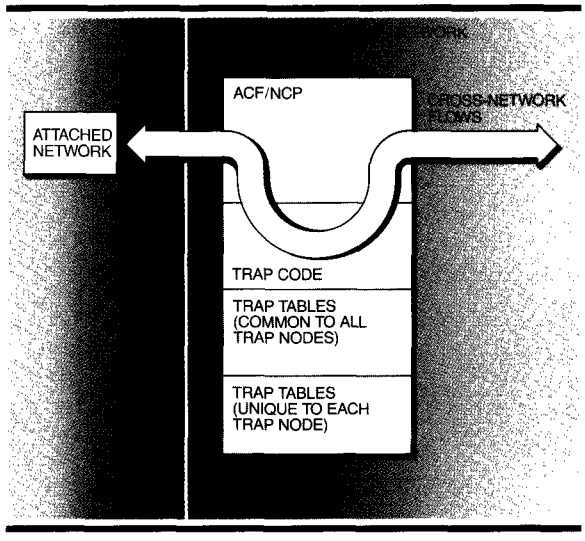


Figure 6 Structure of TRAP Node



work can retain its own network address structure, and each network can manage its own pool of network addresses.

Each network interconnected through TRAP must possess one or more CDRMs. In the TRAP environment, cross-network flows are also cross-domain flows. Figure 3 illustrates both aspects of this communication.

The interconnected network environment includes two basic network types. A single *backbone network* supports the TRAP capability. Multiple *attached networks* are logically and physically connected to the common backbone. The connection points are backbone network communications controllers referred to as *TRAP Nodes*.

A cross-network flow from the backbone network to an attached network is *outbound*. A cross-network flow from an attached network to the backbone network is *inbound*.

An overview of the TRAP multinetwork environment is provided by Figure 4.

TRAP Nodes. A TRAP Node is a backbone network communications controller (an IBM 3705) that contains TRAP code and tables. It is the physical attachment point for one or more attached networks. In addition to its TRAP-related functions shown in Figure 5, a TRAP Node provides normal ACF support. A TRAP Node provides boundary network node support for directly attached devices. It also provides intermediate network node support for intranetwork flows transiting that node. A backbone network may contain one or more TRAP Nodes.

TRAP functions are implemented through extensions to the standard ACF/NCP. This control program is modified at selected points to detect cross-network traffic. When such traffic is encountered, control is passed to special TRAP routines which perform the necessary address transformations.

Address translation information is maintained in tabular form. Some of these tables are common to all TRAP Nodes within the backbone network. Others are unique to a particular TRAP Node. The structure of a TRAP Node is illustrated in Figure 6.

Pseudo Nodes/Links. The TRAP function causes the backbone network and attached networks to perceive a physical and logical environment that is different from the actual configuration. This difference in perception leads to the *Pseudo Node* concept. A Pseudo Node appears as a communications controller interposed between one or more attached networks and a TRAP Node.

Figure 7 Physical view of network interconnection

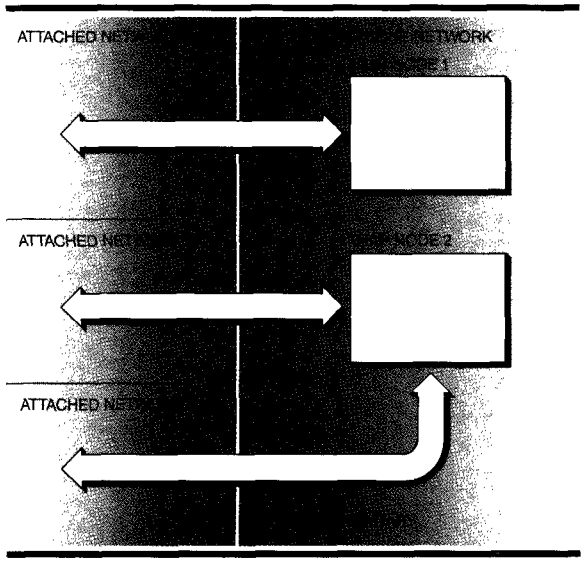
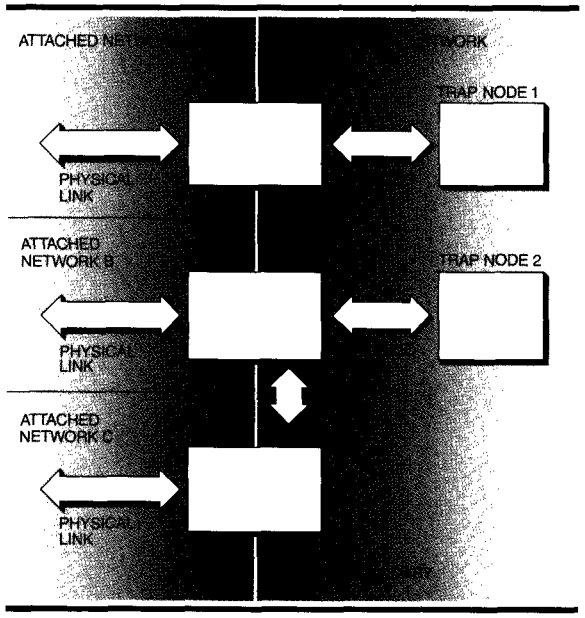


Figure 8 Pseudo view of network interconnection



A Pseudo Node manifests itself to each attached network as part of the address space of that attached network. To the backbone network, a Pseudo Node appears as part of the backbone network address space. Each TRAP Node has one or more backbone network Pseudo Nodes associated with it.

Physical connectivity between a TRAP Node and each attached network consists of one or more links. To the backbone network, all connectivity outboard of a TRAP Node is represented by a single *Pseudo Link*. This Pseudo Link provides a single link appearance for purposes of outboard traffic routing.

Although attached networks perceive Pseudo Nodes, they have no need for a Pseudo Link view of physical connectivity. Each communications controller node in an attached network is limited to a single physical connection to the backbone network. Hence, for attached networks, there is a complete correspondence between perceived and actual links.

When an attached network contains multiple communications controller nodes, it may be possible for more than one such node to be physically connected to the backbone network. In the IBM Information Network environment, most attached networks will be connected to the backbone network via a single attached network communications controller node and hence via a single physical link. Subsequent discussions and illustrations will emphasize this latter environment.

Figures 7 and 8 compare the physical characteristics of an interconnected network environment with the Pseudo Node/Pseudo Link view of that same environment. In Figure 7, three attached networks are connected to the backbone network. Each attached network has a single cross-network link. Two TRAP Nodes provide connection points to the backbone network.

As shown in Figure 8, TRAP alters the appearance of the cross-network environment. One or more Pseudo Nodes appears to be outboard of each TRAP Node in the backbone network.

For attached networks and for the backbone network, each Pseudo Node provides a representation or mapping of all cross-network resources accessible through a particular TRAP Node. To the backbone network, all cross-network resources within attached networks appear to reside in Pseudo Nodes. Likewise, all cross-network resources within the backbone network are represented to attached networks as resources within Pseudo Nodes.

A single Pseudo Node, as viewed by the backbone network, can provide a mapping of one or more attached networks. Additional Pseudo Nodes can be introduced to provide additional mapping capability.

An attached network may view one or more Pseudo Nodes, as determined by the number of cross-network resources to be accessed and by the mapping capability of each Pseudo Node.

SNA message units. TRAP supports the multinetwork environment by performing address transformations as messages transit a network boundary. The message units processed by TRAP are *Path Information Units* (PIUs). Each PIU consists of a *Transmission Header* (TH), a *Request/Response Header* (RH), and a *Request/Response Unit* (RU). Figure 9 illustrates the internal structure of a PIU.

Pseudo Nodes are a mapping device for representing cross-network resources.

The TH contains an *Origin Address Field* (OAF) and a *Destination Address Field* (DAF). The TH serves to route the PIU to its correct destination.

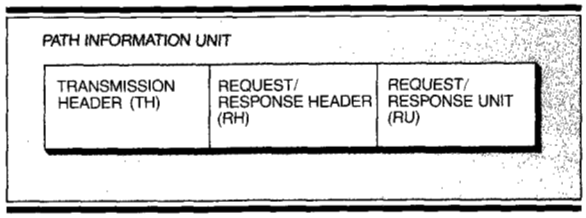
The RH provides control information that identifies the RU as either a request or a response and provides further details about the RU.

The RU is the data portion of the transmission. Although it has no direct role in routing the PIU, the RU may contain imbedded network addresses.

Network environment. The preceding concepts and terminology can now be used to describe the overall TRAP network environment.

An attached network is physically connected to the backbone network via a communications link between a TRAP Node and a communications controller in the attached network. It is possible for a given attached network to have multiple physical connections to the backbone network. However, any given communications controller in an attached network may only have one such connection.

Figure 9 PIU structure



Various other constraints and considerations apply to the definition of a TRAP configuration, but these are not essential to a basic discussion of the cross-network environment.

For each TRAP Node, it is necessary to allocate one or more backbone network subareas to Pseudo Nodes accessible through that TRAP Node. The required number of Pseudo Node subareas is determined by two factors:

- Number of elements per backbone network subarea, as determined by the backbone network address split
- Number of attached network resources to be accessible through a given TRAP Node

For example, if the backbone network address structure allows 1024 elements per subarea, a single Pseudo Node subarea will allow cross-network access to that number of attached network resources. Additional Pseudo Node subareas can be allocated from the backbone network address space as needed. Each additional Pseudo Node subarea would provide access to an additional 1024 cross-network resources in attached networks.

Pseudo Nodes are a mapping device for representing cross-network resources. They are not associated with specific attached networks. The mapping capability provided by a single Pseudo Node may represent cross-network resources in multiple attached networks.

The backbone network subareas allocated to Pseudo Nodes are unique. Since different TRAP Nodes have access to disjoint sets of attached network resources,

different backbone network subareas must be allocated to the Pseudo Nodes accessible through those TRAP Nodes. Otherwise, the backbone network would view different attached network resources as having the same Pseudo Node addresses.

We provide specific examples of backbone network Pseudo Node usage later in the paper.

For each attached network, it is necessary to allocate one or more attached network subareas to Pseudo Node representations of backbone network resources. These subareas are allocated from the address space of each attached network and have no relationship to Pseudo Node subareas allocated in the backbone network or in other attached networks. The two factors determining the required number of Pseudo Node subareas are

- Number of elements per attached network subarea, as determined by the attached network address split
- Total number of cross-network resources in the backbone network

All cross-network resources in the backbone network are addressable at each point where an attached network is connected. Each attached network allocates its own unique set of attached network subareas to access these backbone network resources. For each attached network, these backbone network resources are assigned the same attached network addresses at each connection point. This mapping technique ensures that a backbone network resource has but a single address representation to a given attached network.

Specific examples of attached network Pseudo Node usage are provided later.

Sample network environment. In Figures 10 and 11, a sample multinetwork configuration is used to illustrate relevant features of the TRAP environment. The multinetwork configuration is viewed first from a physical standpoint and then from an apparent, or pseudo, standpoint.

In Figure 10, two attached networks are connected to the backbone network. The backbone network

Figure 10 Physical view of sample network environment

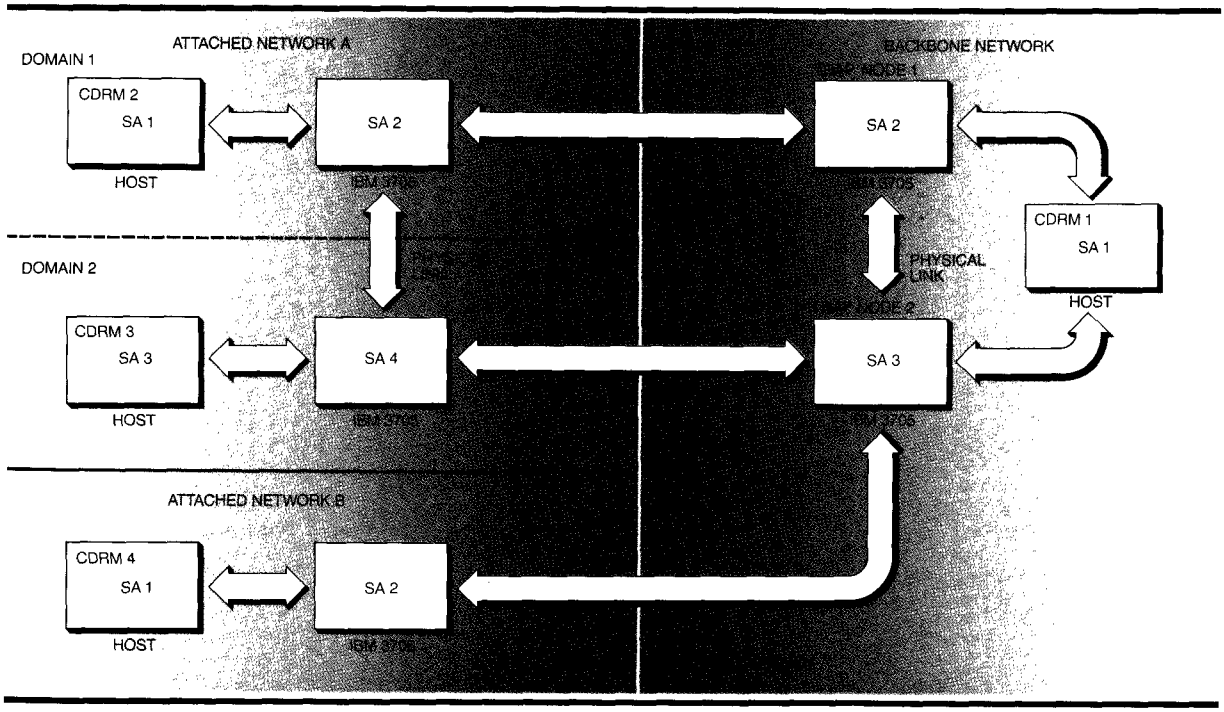
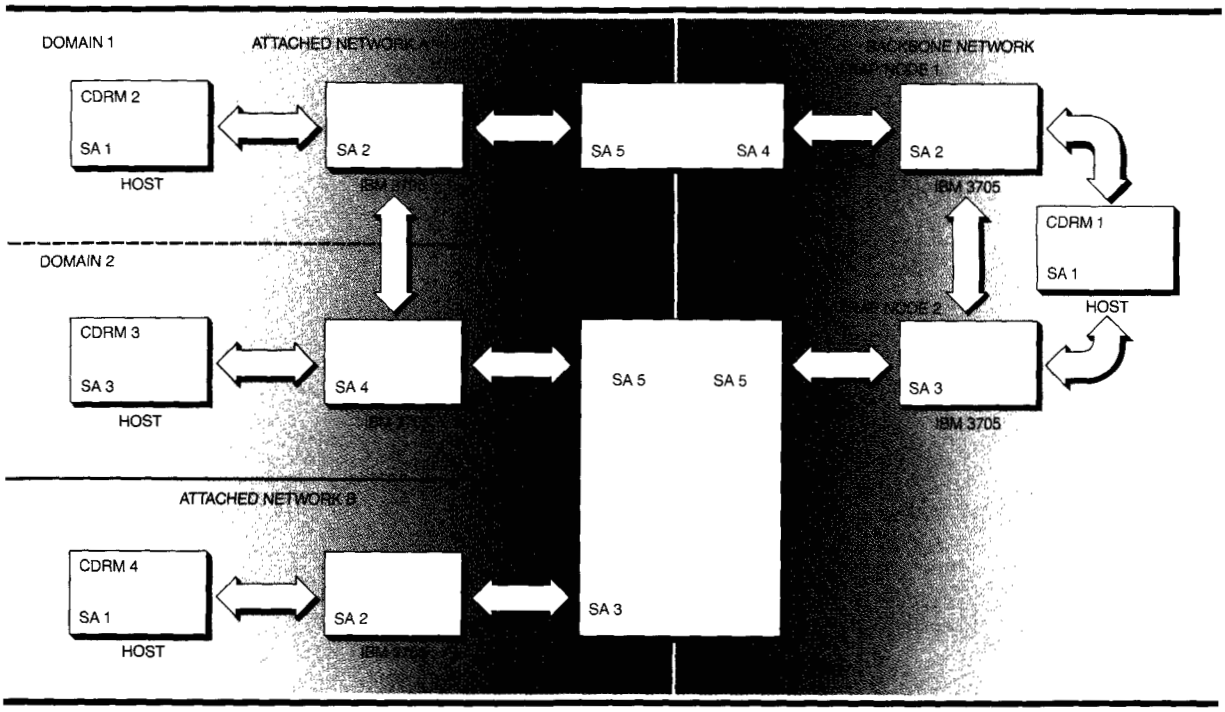


Figure 11 Pseudo view of sample network environment



contains two TRAP Nodes, and these TRAP Nodes support a total of three cross-network physical links.

Attached network A contains two domains. Attached network B is a single-domain network. The backbone network also consists of a single domain.

The network addressing structure and the CDRM functions may be considered part of the logical environment, rather than components of the physical environment. However, they are included in this figure to illustrate potential difficulties in multidomain configurations.

There are a total of four domains in Figure 10. This multidomain environment is managed through cross-domain communication between CDRM pairs. Normal use of the ACF Multi-System Networking Facility would require that all domains utilize the same subarea/element split for network addresses. Furthermore, all subarea usage throughout the entire network would be unique. Otherwise, correct routing of messages would be impossible.

The sample environment clearly violates the requirement for uniqueness of subarea usage. Within its domain(s), each network has chosen to assign subareas sequentially, starting with subarea 1 (SA 1). This technique has produced extensive overlapping of subarea usage. The usage may be reasonable within a given network, but standard MSNF integration of the networks produces a completely unworkable configuration.

The subarea/element split used by each network is not shown. If these network address structures were different, still another barrier to communication would exist.

TRAP Nodes modify the apparent cross-network connectivity. These modifications allow each network to select its own network address subarea/element split and to define its own network address usage.

Figure 11 illustrates the pseudo view of the sample network environment. The TRAP Nodes have caused the appearance of Pseudo Nodes at the boundary between backbone and attached networks. These

Pseudo Nodes allow each network to perceive one (or more) additional nodes within the address space of that network. The Pseudo Node conforms to the network address split and subarea usage of the network in which it appears. The Pseudo Node provides a representation of cross-network resources in a form acceptable to the viewer.

Physical connectivity across network boundaries is unaffected by TRAP. For routing purposes, however, the backbone network perceives a Pseudo Link rather than the multiple physical links that may exist. Since attached network nodes only have a single link to the backbone network, they have no need for the Pseudo Link concept. For attached network nodes, there is a complete correspondence between the perceived link and the actual physical link.

The backbone network accesses different sets of cross-network resources through different TRAP Nodes. Therefore, different backbone subarea addresses must be allocated to the respective Pseudo Nodes. Backbone network subarea 4 (SA 4) represents a Pseudo Node that provides access to the cross-network resources in domain 1 of attached network A. Backbone network subarea 5 (SA 5) represents a Pseudo Node providing access to cross-network resources in domain 2 of attached network A and to cross-network resources of attached network B.

Each attached network accesses the full set of cross-network resources at each point of connectivity. Since the same set of backbone network resources is accessible through each connection, the same subarea addresses must be allocated to the Pseudo Nodes viewed by a given attached network. Therefore, attached network A has two paths to the resources that appear to reside in attached network subarea 5. This subarea represents a Pseudo Node that provides access to cross-network resources in the backbone network.

Attached network B views the same set of cross-network resources, but these resources are represented via a Pseudo Node in the address space of that network. Subarea 3 has been chosen to represent the Pseudo Node in attached network B.

The total number of cross-network resources that can be viewed through a single Pseudo Node is dependent on the network address structure of the network in which the Pseudo Node appears. All the element addresses associated with the Pseudo Node subarea are available for representing cross-network resources. In the example being discussed, each network views a set of cross-network resources that can be represented by a single Pseudo Node subarea. If additional cross-network resources exist, additional Pseudo Node subareas may be assigned.

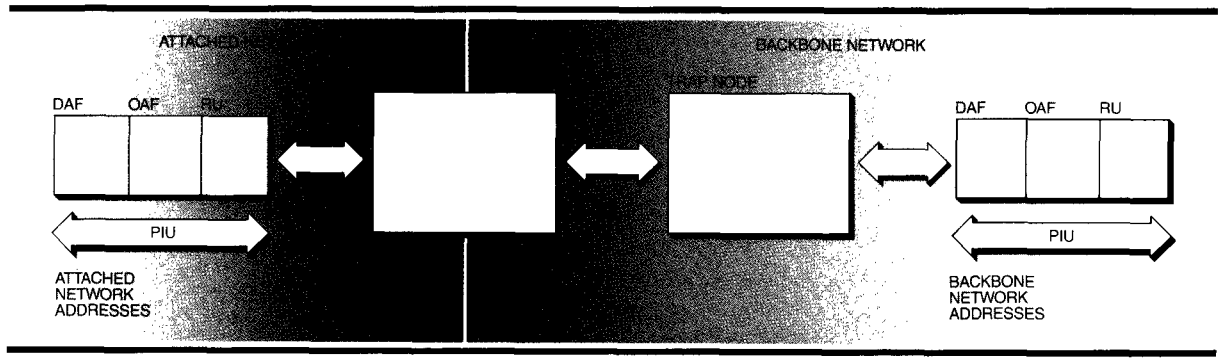
TRAP Nodes facilitate communication across the network boundary between the backbone network and the attached networks. TRAP Nodes do not directly affect the network boundaries between attached networks. In Figure 11, attached network A and attached network B cannot communicate directly. They have no physical connectivity, they may have different network address structures, and they have overlapping subarea assignments.

Network address translation is the fundamental TRAP process.

The attached networks may, however, communicate via the backbone network. The cross-network resources of domain 1 in attached network A are known to the backbone network as resources within backbone network subarea 4. These subarea 4 backbone resources can then be defined to attached network B as resources within subarea 3 of that network. Attached network B would thereby have complete addressability and routing capability to domain 1 of attached network A. Similar definition of cross-network resources to attached network A would provide connectivity in the reverse direction.

TRAP has allowed address space independence in the network environment shown above. However, no name space independence has been provided. All cross-network resources must have unique names. Furthermore, each network must ensure that names within that network do not conflict with cross-network names known to that network. CDRMs, by definition, are cross-network resources. Therefore, the CDRMs shown above all have unique names.

Figure 12 TRAP address translation



Processing. The basic characteristic of the TRAP network environment is the address space isolation between interconnected networks. This isolation is achieved through TRAP address transformations performed at the network boundary. Accordingly, network address translation is the fundamental TRAP process.

A TRAP Node detects cross-network Path Information Units (PIUs) and translates associated network addresses. These addresses always include the Origin Address Field (OAF) and the Destination Address Field (DAF) for the PIU. For certain communications between CDRMs, the Request/Response Unit (RU) within the PIU is also examined for imbedded network addresses. Figure 12 depicts the TRAP address translation.

The address translation process converts addresses that are valid in the originating network into addresses that are valid within the destination network. Resources within the originating network are assigned the subarea address of a Pseudo Node within the destination network.

Network addresses imbedded within RUs typically represent resources managed by a pair of CDRMs. The addresses of cross-network resources are dynamically exchanged between pairs of CDRMs as part of the session establishment process. Translation of these imbedded addresses allows a cross-network resource to appear as a resource within the same address space as the receiving CDRM.

For outbound transmissions, a TRAP Node detects cross-network flows by examining the DAF in the PIU. If the DAF represents a Pseudo Node subarea outboard of a given TRAP Node, address translation is required by that TRAP Node.

For inbound transmissions, a TRAP Node can detect a cross-network flow by reference to the physical link over which the PIU was received.

The TRAP function does not require TRAP Nodes to be session end points for cross-network sessions. Accordingly, TRAP Nodes have little sensitivity to the nature or status of such sessions. For normal session traffic, TRAP Node processing is limited to PIU address translation.

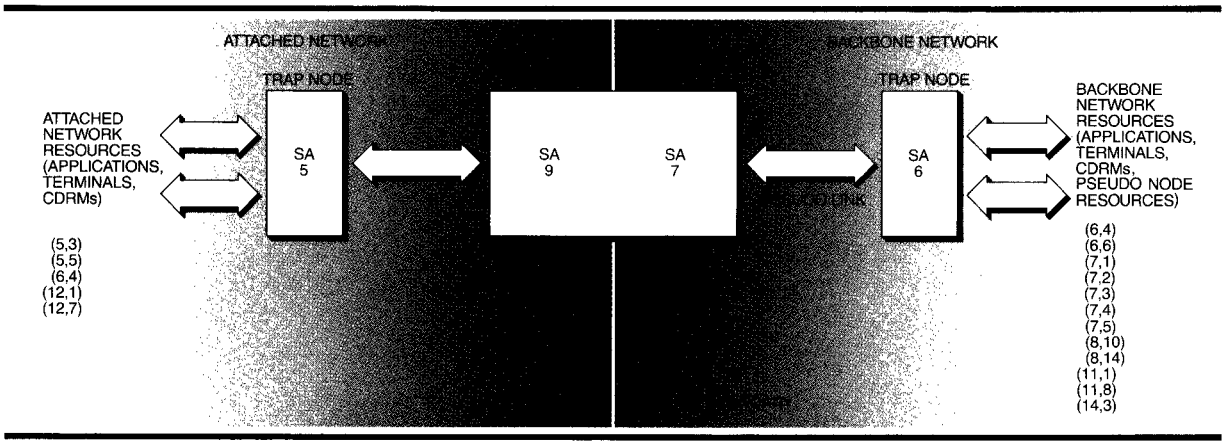
TRAP Nodes do maintain a mimimal session awareness. This awareness includes

- Session initiation/termination
- Session type (e.g., CDRM ↔ CDRM)

A TRAP Node uses this session information in response to certain network error conditions. The session data allow the TRAP Node to determine which session end points should be notified and what form of notification should be used.

When an error occurs within an attached network or within the backbone network, there may be an attempt to report this error across the network boundary. For example, if a node in the attached network experiences an outage, the loss of the associated subarea may be reported over the cross-network link. This outage notification cannot be propagated directly into the backbone network, since it refers to a subarea in a different address space. Accordingly, the TRAP Node must intercept such notifications and convert any address references to backbone network equivalents.

Figure 13 Sample Cross-Network Configuration I



Similar considerations apply when the backbone network attempts to report errors across the network boundary. The TRAP Node will intercept these notifications and convert them to a form usable by the attached network(s).

The address translation tables required by TRAP are generated by a special utility program. This program is run off-line as a normal batch job and produces a complete and customized set of tables for each TRAP Node. These tables can then be link-edited with the control program for each TRAP Node. Updating of translation table information requires reloading of the TRAP Node, but it does not require regeneration of the ACF/NCP for that node.

TRAP address translation

As noted earlier, network address translation is the fundamental TRAP process. This process can best be discussed in the context of specific network examples. The TRAP algorithm will be described in sufficient detail to illustrate its basic function. Detailed design and implementation information is omitted.

Session management, error handling, and various other functions are also omitted from the following discussion. Though important, they are peripheral to the basic TRAP processes.

Tables. Several information tables are necessary within each TRAP Node to perform cross-network address translations. Two of the most significant are described below. The relationship of these tables to the actual translation process will be illustrated in subsequent discussions of specific cross-network environments.

The Attached Network Table contains the backbone network addresses of all backbone network resources that are to be known cross-network. This table is sorted into ascending order.

Since the complete set of backbone network cross-network resources is accessible through each TRAP Node, all TRAP Nodes contain an identical Attached Network Table.

The TRAP Node Table contains the attached network addresses of all attached network resources that are to be known cross-network and that are accessible via a particular TRAP Node. This table is also sorted into ascending order.

Since each TRAP Node provides access to a unique set of attached network resources, each TRAP Node contains a unique TRAP Node Table.

Cross-Network Configuration I. The cross-network configuration in Figure 13 shows a backbone network and a single attached network. Details of the attached and backbone network configurations are omitted, except as they apply to representation of the cross-network environment. In this cross-network configuration, the attached and backbone networks are connected via a single pair of nodes. Within the attached network, the attachment point is the communications controller node represented by subarea 5 (SA 5). Within the backbone network,

the TRAP Node has been assigned subarea 6 (SA 6). The physical connection between the two networks is constrained to a single link, since TRAP allows only one cross-network connection between any pair of nodes. In this simple example, the Pseudo Link perceived by the backbone network can be mapped directly to a physical link.

The attached network has chosen subarea 9 to be used to map its view of cross-network resources. Accordingly, this subarea has been assigned to the Pseudo Node seen by the attached network. All cross-network resources known to the attached network will appear to reside in subarea 9. The backbone network has selected subarea 7 for its Pseudo Node representation. All cross-network resources known to the attached network will appear to reside in subarea 7. In this example, the total number of cross-network resources viewed by either network is sufficiently small to require only a single Pseudo Node subarea in each network.

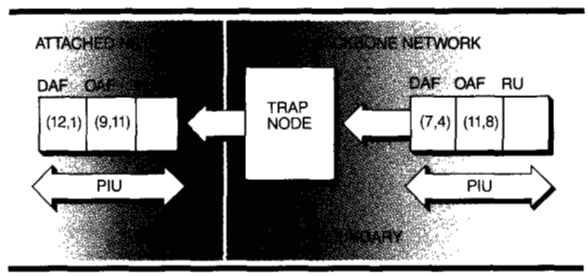
The cross-network resources that reside in the attached network consist of application programs, terminals, and CDRMs. These resources are identified by network addresses in the form (x,y), where the subarea address is represented by x and the element address by y. Thus, the cross-network resources in the attached network are associated with subareas 5, 6, and 12.

Cross-network resources within the backbone network are represented in the same manner. As noted earlier, backbone network Pseudo Node addresses are assigned to represent the cross-network resources in the attached network. These are the Pseudo Node addresses (7,1) through (7,5). These addresses within the backbone network address space may also be considered as cross-network resources owned by the backbone network. This approach allows attached networks to use the backbone network as a medium for attached-network-to-attached-network communication.

The address translation process can be illustrated by referencing the preceding network configuration and the corresponding table definitions. Address translation will be described for both outbound and inbound flows.

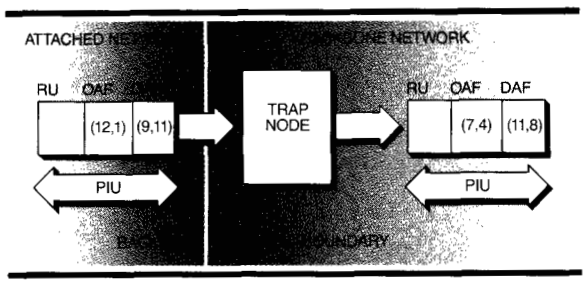
Outbound processing. When a TRAP Node receives a PIU over a backbone network link, the DAF subarea is tested. A sample is shown in Figure 14. If this subarea represents a Pseudo Node outboard of that TRAP Node, address translation proceeds as follows:

Figure 14 Sample outbound cross-network address translation



1. The subarea address 7 and element address (4) are extracted from the DAF.
2. The subarea and element addresses are used to calculate an index into the TRAP Node Table.
3. The attached network address (12,1) is extracted from the indicated TRAP Node Table entry. This represents the attached network address of the attached network resource identified by the original DAF.
4. The DAF field of the PIU is updated with this translated address. The DAF continues to be associated with the same attached network resource. However, the network address has been converted from backbone network form to attached network form.
5. The OAF is extracted from the PIU and used as a search argument for the Attached Network Table.
6. A binary search of the Attached Network Table locates a matching backbone network address.
7. The displacement of this matching backbone network address within the Attached Network Table is used to calculate an index value. This index value is further manipulated to produce an attached network subarea and element address value. In this example, the calculated attached network address is (9,11). This represents the attached network address of the backbone network resource identified by the original OAF.
8. The OAF field of the PIU is updated with this translated address. The OAF continues to be associated with the same backbone network resource. However, the network address has been converted from backbone network form to attached network form.

Figure 15 Sample inbound cross-network address translation



9. Any imbedded RU addresses would be processed by the same mechanisms. Either the OAF or the DAF translation technique would be used. The imbedded RU address would be translated as an OAF if it represented a backbone network resource. Otherwise, it would be translated as a DAF.
10. The PIU is then queued for transmission on the link from the backbone network to the attached network.

Inbound processing. A sample of inbound translation is shown in Figure 15. When a TRAP Node receives a PIU over an attached network link, the PIU is processed as follows:

1. The subarea address 9 and element address 11 are extracted from the DAF.
2. The subarea and element addresses are used to calculate an index into the Attached Network Table.
3. The backbone network address (11,8) is extracted from the indicated Attached Network Table entry. This represents the backbone network address of the backbone network resource identified by the original DAF.
4. The DAF field of the PIU is updated with this translated address. The DAF continues to be associated with the same backbone network resource. However, the network address has been converted from attached network form to backbone network form.
5. The OAF is extracted from the PIU and used as a search argument for the TRAP Node Table.
6. A binary search of the TRAP Node Table locates a matching attached network address.
7. The displacement of this matching attached network address within the TRAP Node Table is used to calculate an index value. This index value is further manipulated to produce a backbone network subarea and element address value. In this example, the calculated backbone network address is (7,4). This represents the backbone network address of the attached network resource identified by the original OAF.
8. The OAF field of the PIU is updated with this translated address. The OAF continues to be associated with the same attached network resource. However, the network address has been converted from attached network form to backbone network form.
9. Any imbedded RU addresses would be processed by the same mechanisms. Either the OAF or the DAF translation technique would be used. The imbedded RU address would be translated as an OAF if it represented an attached network resource. Otherwise, it would be translated as a DAF.
10. The translated PIU is then queued for transmission within the backbone network.

Cross-Network Configuration II. The cross-network configuration in Figure 16 shows a backbone network and two attached networks. Details of the attached and backbone network configurations are omitted, except as they apply to the representation of the cross-network environment. The primary intent of this network example is to illustrate a more complex TRAP environment. Address translation will be described for flows between CDRM 1 in attached network A and CDRM 2 in attached network B. The backbone network will be a transit medium for this communication.

In this cross-network configuration, it is assumed that each attached network connects to the backbone network via a single physical link. Thus, there is a complete equivalence between physical links and Pseudo Links in this example.

Each attached network can communicate with cross-network resources in the backbone network. Each attached network can also communicate with cross-network resources in the other attached network via the backbone network.

Attached network A has chosen subarea 9 for its Pseudo Node view of cross-network resources. All cross-network resources known to attached network A will appear to reside in this subarea. This is true for cross-network resources within the backbone network and within attached network B.

Attached network B has chosen subarea 13 for its Pseudo Node view of cross-network resources. All cross-network resources known to attached network B will appear to reside in this subarea. This is true for cross-network resources within the backbone network and within attached network A.

The backbone network has selected subarea 4 for the Pseudo Node that appears outboard of the subarea 2 TRAP Node. Thus, all cross-network resources in attached network A appear to reside in subarea 4. The Pseudo Node outboard of the subarea 1 TRAP Node has been assigned subarea 3 and provides addressability to cross-network resources in attached network B.

The backbone network view of an attached network is itself identified as a cross-network resource to all attached networks. It is this mechanism that allows attached networks to communicate with one another. The following discussion provides an overview of attached-network-to-attached-network communication.

The translation example will involve a flow from CDRM 1 in attached network A to CDRM 2 in attached network B.

Inbound processing. Following are the steps for inbound processing for the example of Figure 17:

1. When attached network A sends a PIU to attached network B, that transmission is initially treated as an inbound message by the subarea 2 TRAP Node.
2. As described in Cross-Network Configuration I, the DAF and OAF are extracted from the PIU.
3. The DAF is used in conjunction with the Attached Network Table to determine the backbone network address for CDRM 2 in attached network B.
4. The OAF is used in conjunction with the TRAP Node Table to determine the backbone network address for CDRM 1 in attached network A.
5. This processing yields a value of (3,1) for the DAF and (4,1) for the OAF. The appropriate PIU fields are updated with these backbone network addresses.
6. Any imbedded RU addresses are also translated. If such an address refers to an attached-network-A resource, it is translated as for an inbound OAF. If an imbedded RU address represents an attached-network-B resource, it is translated as for an inbound DAF.

Figure 16 Sample Cross-Network Configuration II

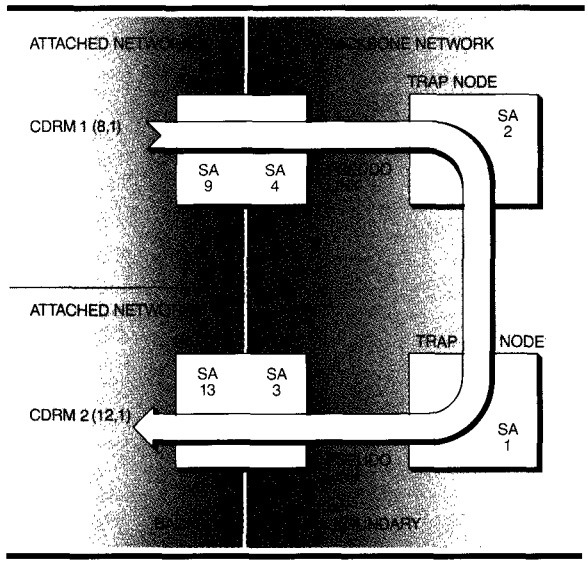
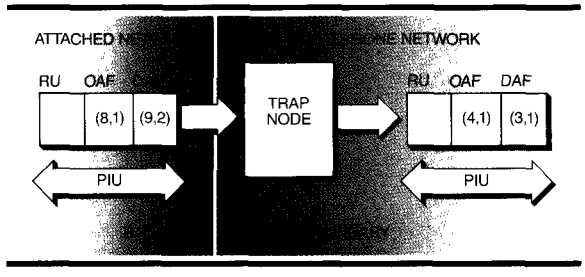


Figure 17 Sample inbound cross-network address translation

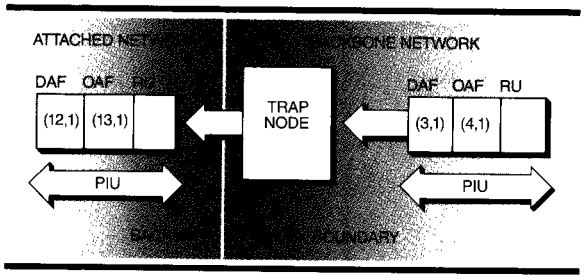


7. The PIU is then queued for transmission within the backbone network to the subarea 1 TRAP Node.

Outbound processing. Following are the outbound processing steps for the example of Figure 18:

1. When the PIU from attached network A reaches the subarea 1 TRAP Node, the transmission is treated as an outbound message unit.
2. As described in Cross-Network Configuration I, the DAF and OAF are extracted from the PIU.
3. The DAF is used in conjunction with the TRAP Node Table to determine the attached network B address for CDRM 2 in attached network B.

Figure 18 Sample outbound cross-network address translation



4. The OAF is used in conjunction with the Attached Network Table to determine the attached network B address for CDRM 1 in attached network A.
5. This processing yields a value of (12,1) for the DAF and (13,1) for the OAF. The appropriate PIU fields are updated with these attached network B addresses.
6. Any imbedded RU addresses are also translated. If such an address refers to an attached network A resource, it is translated as for an outbound OAF. If an imbedded RU address represents an attached network B resource, it is translated as for an outbound DAF.
7. The PIU is then queued for transmission to attached network B.

TRAP operational considerations

The operational characteristics of a TRAP environment are not greatly different from a standard MSNF environment. Cross-network resources are represented with translated addresses, but they otherwise appear as normal cross-domain resources. Cross-domain communication and protocols across a network boundary are not affected by TRAP.

Resource ownership cannot be transferred across a network boundary. Although resources can be shifted between domains within the same MSNF network, this is not possible when the domains reside in separate networks. The cross-network view of resources and configurations is neither complete enough nor accurate enough to allow such resource transfer.

Problem determination in a TRAP environment is complicated by the lack of an end-to-end view of sessions and flows. All flows appear to terminate within the address space of each network. Each domain within a network may perform normal problem determination activity for the resources that it manages. An end-to-end perspective requires the coordination of these individual domain activities and the reconciliation of multiple address structures.

For backbone network routing purposes, cross-network links attached to a given TRAP Node are viewed as a single Pseudo Link. For purposes of activation, deactivation, and problem determination, these physical links appear as normal cross-domain links to network operators in both the attached and backbone networks.

Cross-network flows must be controlled to ensure that network addresses are not exchanged as data between users. TRAP will translate any imbedded RU addresses defined by SNA. If end users include network addresses as part of an unformatted data stream, TRAP will not detect these addresses and will be unable to provide the necessary translation. Since all architecturally defined RUs are processed properly, this constraint is unlikely to have any practical effect on network operation.

Some ACF functions are prevented from operating across a TRAP boundary. A detailed discussion of these restricted functions is beyond the scope of this paper. The functional limitations stem from implementation resource and schedule considerations. TRAP has no intrinsic design limitations on full ACF function. Flows and functions contained entirely within an attached or backbone network are not restricted by TRAP.

All TRAP functions, flows, and configurations discussed in this paper are fully achievable with the basic TRAP implementation.

TRAP performance characteristics

Common performance parameters include storage requirements, execution cycle requirements, and response times. The TRAP implementation has been carefully analyzed from these standpoints.

TRAP depends on statically defined tables in achieving its address translation functions. These table structures can become significant in size. Although a detailed explanation of TRAP storage usage is beyond the scope of this paper, a general storage requirement can be defined. For large cross-

network environments involving multiple TRAP Nodes, multiple attached networks, and several thousand terminals, 50K to 60K bytes of communications controller storage is adequate for both TRAP code and tables. In the IBM Information Network environment, this increased storage use has been acceptable.

TRAP depends on statically defined tables in achieving its address translation function.

Detailed analysis of the overhead required for address translation had indicated a negligible impact on communications controller utilization. Although TRAP performs some additional processing for session initiation/termination and for certain network error conditions, the basic TRAP functions add little to communications controller processing.

TRAP has undergone extensive testing with the Teleprocessing Network Simulator (TPNS). This testing involved a multiplicity of links, link protocols (Synchronous Data Link Control, or SDLC, and binary synchronous communication, or BSC), terminal types, transaction profiles, and applications. No perceptible impact on end user response time was introduced by the address translation process.

TRAP production usage

The first TRAP cross-network flows were achieved in April 1982. Since that time, TRAP has undergone extensive testing with a wide variety of hardware and software combinations in the backbone and attached networks.

TRAP was phased into full production in the IBM Information Network during August and September 1982. This production usage required TRAP to support interconnection of the Information Network development and production networks. In this environment, the development network assumes the role of an attached network, and the production network serves as the backbone network. Relative to using normal MSNF connectivity, the TRAP approach has provided a net savings of approximately 25 percent of the backbone network address space.

In addition to internal use of TRAP by the Information Network, all external services and offerings that involve MSNF attachment of user SNA networks are based on TRAP. Current examples are the Network Services and Insurance Communications Service offerings.

Summary

The integration of SNA networks becomes more frequent as such networks increase in size and number. This integration provides enhanced opportunities for sharing resources among networks.

When SNA networks are merged through the standard ACF Multisystems Networking Facility, these networks effectively become a single entity. All portions of this new network must share the use of certain resources. One critical resource is the pool of network addresses. Networks that are joined through the standard ACF MSNF capabilities must share this finite quantity of network addresses and must observe common usage standards for those addresses.

TRAP is an experimental technique that alleviates some difficulties in the multiple-network environment. It is an algorithm that has been developed by the IBM Information Network to meet its specific needs. Although TRAP is a limited solution, it illustrates one means to achieving address space isolation between networks. TRAP also provides a reference point for investigating and evaluating more sophisticated approaches to interconnected networks, such as the recently announced SNA Network Interconnection function.

This paper has described both the conceptual basis and the implementation of the TRAP algorithm. TRAP causes interconnected networks to perceive an environment that is substantially different from the actual network configuration. These modifications to cross-network perception allow each network to retain essentially independent network address usage.

TRAP is implemented via extensions to the ACF Network Control Program that operates in the IBM 3705 Communications Controller node. TRAP plays an important role in the interconnection of both internal and external networks and as such has provided a reliable, useful response to the specific requirements of the IBM Information Network.

Acknowledgments

The author wishes to express his gratitude to the TRAP project team for their dedicated and professional efforts. The skill and experience of S. G. Bush and R. L. Naylor were particularly critical to the successful implementation of the TRAP concept.

Appendix: TRAP glossary

Definitions have been provided for significant TRAP-related terminology. These definitions are intended to facilitate an understanding of TRAP concepts. They are not intended to provide rigorous definitions in any broader context.

ACF — Advanced Communications Function is a set of IBM programs that provides communications capabilities based on SNA.

ACF/NCP — ACF/NCP is the Network Control Program that operates in the IBM 3705 Communications Controller. TRAP function is achieved by the addition of code and tables to the standard ACF/NCP.

Attached network — This term refers to any of several networks that may be attached to the backbone network. These networks are not required to have any TRAP sensitivity and communicate with the backbone network using normal MSNF (see backbone network).

Attached Network Table — This term refers to one of several tables required by TRAP Nodes to perform address translation. The attached Network Table contains the backbone network address of all backbone network resources that are to be known across the network boundary (cross-network).

Since each TRAP Node provides access to the complete set of backbone network cross-network resources, all TRAP Nodes have an identical Attached Network Table.

Backbone network — The backbone network is the network owning and controlling the TRAP Nodes (see attached network).

CDRM — The Cross-Domain Resource Manager is the function within an SNA domain that manages resource usage across domain boundaries.

Cross-domain — This adjective pertains to resources or functions that are seen across or that operate across a domain boundary. Use of resources across a domain boundary requires communication between a pair of Cross-Domain Resource Managers. Communication through a TRAP Node is cross-domain communication.

Communications controller — The IBM 3705 Communications Controller is a transmission control unit that provides many of the line control functions required by a communications environment. The control program for this node is ACF/NCP.

Cross-network — This adjective pertains to resources or functions that are seen across a TRAP Node or that operate across a TRAP Node. Communication through a TRAP Node is cross-network communication.

DAF — The Destination Address Field in the TH indicates the network address to which a PIU is to be sent.

Domain — This term refers to the set of logical and physical resources managed by a single resource manager within an SNA network. An SNA network may be composed of multiple domains and may therefore contain multiple resource managers. Interactions between domains require communication between Cross-Domain Resource Managers.

Host — A host consists of a System/370 CPU and its associated ACF access method. The host node is typically the focal point for resource ownership and management within a network domain. The TRAP implementation does not involve any modifications to host nodes.

IBM Information Network — The IBM Information Network is a computing and network complex that allows a wide range of users access to applications, data banks, and productivity tools. This access is primarily through user terminals connected to this SNA-based network.

With TRAP, this connection capability is extended to include user networks. These user networks use the ACF MSNF to communicate with the IBM Information Network. The Network Services and Insurance Communications Service offerings provide attachment based on TRAP.

Inbound — This adjective describes flows that proceed from an attached network to the backbone network.

MSNF — This acronym stands for Multisystems Networking Facility and refers to the standard ACF functions in support of multiple domain environments.

Network — In the context of this paper, a network is a collection of SNA logical and physical resources that are grouped into one or more domains. An SNA network is characterized by a common address space. TRAP allows interconnection of multiple SNA networks while substantially maintaining the independent nature of each network. Thus, communication through a TRAP Node is cross-network communication.

OAF — The Origin Address Field in the TH indicates the network address from which a PIU has been sent.

Outbound — This adjective describes flows that proceed from the backbone network to an attached network.

PIU — The Path Information Unit is a message unit typically consisting of TH, RH, and RU.

Pseudo Link — This term refers to the link that appears to connect a TRAP Node with one or more Pseudo Nodes. There may be multiple attached networks with multiple physical connections to a given TRAP Node. From the standpoint of backbone network routing, these multiple physical links appear as a single Pseudo Link connection to a Pseudo Node.

A single Pseudo Node appears directly accessible via the Pseudo Link. All other Pseudo Nodes accessible from a given TRAP Node appear to be outboard of this adjacent Pseudo Node.

Pseudo Node — This term refers to one or more communications controller nodes that appear to be interposed between the attached network and the backbone network. These nodes do not in fact exist. However, TRAP creates the illusion of additional communications controller nodes, as seen by both the attached and backbone networks.

A Pseudo Node appears to the backbone network as though it were in a separate domain within the backbone network. A Pseudo Node is represented as a normal backbone subarea outboard of a TRAP Node and is subject to standard routing definitions within the backbone network.

The attached network has a similar view of Pseudo Nodes. A Pseudo Node appears to reside in a separate domain within the attached network. Standard routing definitions apply within the attached network.

RH — The Request/Response Header follows the TH in the PIU and contains control information about the request or response. A Request Header (RH) is normally followed by a Request Unit (RU). A Response Header (RH) is optionally followed by a Response Unit.

RU — The Request/Response Unit (RU) is a message unit contained within a PIU and preceded by an RH. A Request Unit (RU) contains information such as a request code, end user data, etc. A Response Unit (RU) contains positive or negative response information related to a previous request.

SNA — Systems Network Architecture is a comprehensive specification of formats and protocols for communications environments.

SNA Network Interconnection — The recently announced, strategic IBM response to the multiple-network environment. This function provides name and address space isolation between communicating networks.

TH — The Transmission Header is the portion of a PIU that provides routing information for message units sent through a network. It precedes the RH and RU.

TPNS — The Teleprocessing Network Simulator is a program product that allows simulation of a wide range of communications environments. TPNS is commonly used to analyze the performance of network components under load.

TRAP — An interim, experimental network interconnection capability implemented by the IBM Information Network. TRAP provides address space isolation between communicating networks.

TRAP Node — This term refers to a communications controller node that provides TRAP function. This function is achieved through the addition of code and tables to the standard ACF/NCP in the communications controller node.

TRAP Node Table — This term refers to one of several tables required by TRAP Nodes to perform address translation. The TRAP Node Table contains the attached network addresses of all attached network resources that are to be known cross-network and that are accessible via a particular TRAP Node.

Since each TRAP node provides access to a unique set of attached network cross-network resources, each TRAP Node has a unique TRAP Node Table.

Cited reference

1. J. H. Benjamin, M. L. Hess, R. A. Weingarten, and W. R. Wheeler, "Interconnecting SNA networks," *IBM Systems Journal* **22,** No. 4, 344–366 (1983, this issue).

General references

Advanced Communications Function: Introduction, GC30-3033, IBM Corporation; available through IBM branch offices.

ACF/NCP/VS (Network Control Program, Systems Support Programs): General Information, GC30-3058, IBM Corporation; available through IBM branch offices.

ACF/VTAM: General Information, GC27-0462, IBM Corporation; available through IBM branch offices.

Systems Network Architecture: Concepts and Products, GC30-3072, IBM Corporation; available through IBM branch offices.

Systems Network Architecture: Format and Protocol References Manual, SC30-3112, IBM Corporation; available through IBM branch offices.

Systems Network Architecture: Reference Summary, GA27-3136, IBM Corporation; available through IBM branch offices.

Systems Network Architecture: Sessions Between Logical Units, GC20-1868, IBM Corporation; available through IBM branch offices.

Systems Network Architecture: Technical Overview, GC30-3073, IBM Corporation; available through IBM branch offices.

Reprint Order No. G321-5200.

Keith D. Ryder *IBM Japan, Work Station Business Unit, Masonic 39 Mori Building, 4-5, Azabudai 2-chome, Minato-Ku, Tokyo 106, Japan.*
 After joining IBM in 1965, Mr. Ryder worked on control program design and development for the OS/360 MFT and MVT systems. Subsequently, he was involved in the definition of the address translation architecture for System/370 virtual systems. He then participated in the architecture, design, and performance analysis of OS/VS2 Release 1. He contributed to the initial definition of Systems Network Architecture and was involved in the overall design of ACF. In 1976 and 1977, he was part of a communications joint study between the American Express Corporation and IBM's former System Communications Division. From 1977 to 1980, he was assigned to the Installation Support Centre in England, providing planning and technical support for ACF installations from IBM's Europe/Middle East/Africa Corporation. From 1980 through mid-1983, he worked in Network Planning and Design for the IBM Information Network in Tampa. At the present time, he is assigned to the Work Station Business Unit in Japan. Mr. Ryder received an M.A. in mathematics from the University of Detroit in 1965.