Managing changes in SNA networks

by C. P. Ballard
L. Farfara
B. J. Heldke

Systems Network Architecture/Management Services (SNA/MS) has been enhanced to give network users change management capabilities. The first IBM products implementing change management are NetView™ Distribution Manager R2 and the 3174 Control Unit with the Central Site Change Management microcode function. This paper describes the design selected and the functions provided: Retrieve, Send, Delete, Install, Send-and-Install, Remove, Accept, and Activate. It also describes how SNA/MS makes use of another new SNA component designed for it—SNA/File Services, described in another paper in this issue. (Although not strictly necessary, it is recommended that the other paper be read prior to reading this one.) SNA/File Services, in turn, uses an enhanced SNA/Distribution Services format to provide an architecture for file distribution in an SNA network.

Since its introduction in 1974, IBM's Systems Network Architecture (SNA) has gained wide market acceptance.[1,2] The appearance of ever-larger SNA networks has created the need for centralized management capabilities, including the ability to assign management focal points (by geographic region, for example) and the ability to handle interconnected but independently administered networks. Increasingly diverse technologies and a multiplicity of vendors in the networking scheme have led naturally to the development of a network management component of SNA to provide these capabilities—SNA/Management Services (SNA/MS).[3–5]

An early requirement of managers responsible for operation of an SNA network was to be able to retrieve files containing both executable objects (for example, programs or panels) and associated data sets from one or more SNA nodes at which the files could be prepared, bring them to a central administrative site, and subsequently distribute them to remote SNA nodes for execution or processing. Also required was the ability to delete files at the remote nodes. Users needed to plan, schedule, and track all of these activities from a central site. The Distributed Systems Executive (DSX) program was developed to meet these requirements for a System/370 central site host and was first released in 1978. Over time, it was enhanced to accommodate a variety of remote systems, such as the IBM 8100, Series/1, System/36, and System/370 with the VSE operating system. More recent versions added support for IBM 4680 Store System Processors and IBM Personal Computers.

By the mid-1980s, the need to combine several network management products into one overall network management product strategy was recognized. This strategy led IBM to group a number of licensed programs for network management into the NetView™ product family with the aim of facilitating the integration of functions and user interfaces. In October 1987, IBM announced the NetView Distribution Manager[6] product that is based on DSX extensions and improvements. It adds support for IBM Application System/400™ (AS/400™), System/370 nodes with the Virtual Machine/System Product (VM/SP) or Distributed Processing Programming Executive (DPPX) operating systems, Personal System/2®, and System/88.

As the variety of remote SNA nodes supported by the NetView Distribution Manager increased, requirements for architecture to support its functions became apparent. These requirements were in three major areas:

- Change management
- System-independent file management and distribution
- Bulk data transport (including store-and-forward and fan-out features)

In the case of change management, the architecture was required to support the wide variety of products participating in an SNA network and to be open so that users and other vendors could implement it if desired. SNA/MS enhancements were developed to provide the change management functions, and this paper explores the architectural solution for these functions in more detail. SNA/MS makes use of a new SNA component, SNA/File Services (SNA/FS), that was developed to address the file distribution functions. Another paper in this issue[7] and a reference manual[8] describe SNA/FS in detail. The existing SNA/Distribution Services (SNA/DS),[9,10] with streamlined formats, was chosen by SNA/FS to satisfy the requirements for bulk data transport. The Logical Unit Type 6.2 (LU 6.2) protocol is used to exchange data between nodes participating in the SNA/DS network.

The first implementing products are NetView Distribution Manager R2 and the IBM 3174 Control Unit with the Central Site Change Management microcode function.

Change management roles and responsibilities

SNA/MS defines an *entry point* as an SNA node that sends network management data (in SNA/MS format) about itself and the resources it controls to a focal point for centralized processing, and that receives and executes focal-point-initiated commands to manage and control its resources. A *focal point* is an entry point that accepts specific management and control requests of a particular network management category (for example, change management) from a user of SNA/MS (arbitrarily termed *the network planner* for architectural modeling convenience) and issues corresponding commands to other entry points.

SNA/FS capability is symmetric with respect to focal point and entry point roles.

The user has the opportunity to centralize management and control at one or more focal points.

SNA/FS capability is symmetric with respect to focal point and entry point roles. That is, nodes in either role can send files to, and retrieve files from, other nodes. Change management commands, however, are only issued from a focal point. Figure 1 shows a network of nodes participating in change management.

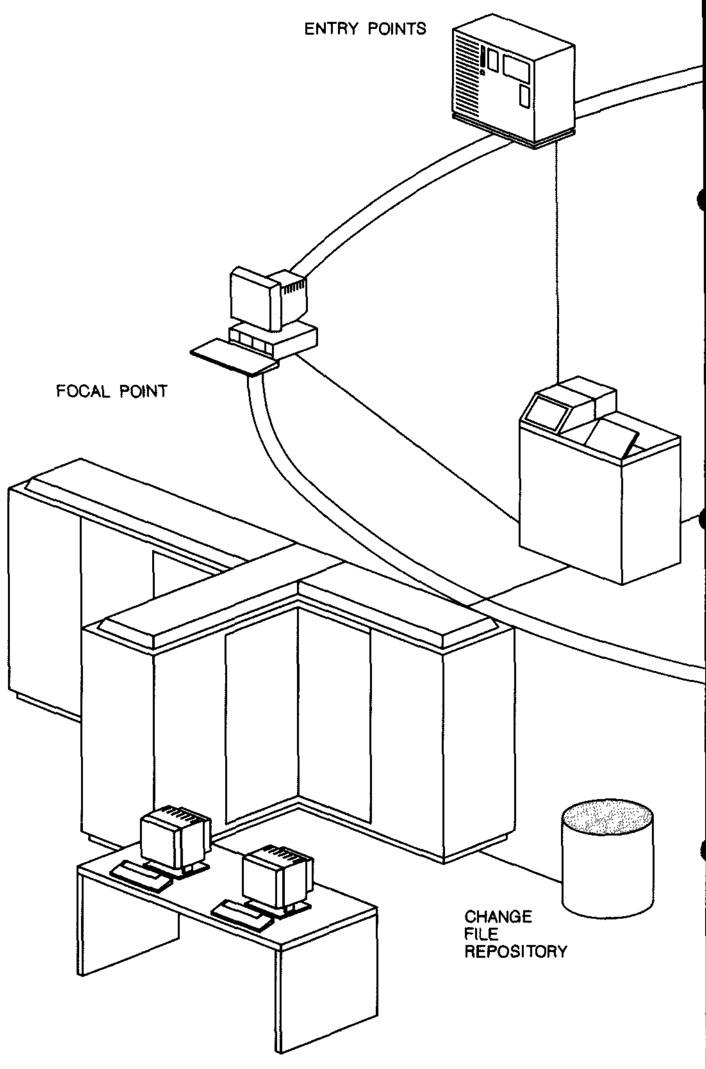
Entry points are typically remote and unattended. An entry point may be a large system or a small one, an intelligent workstation, a fixed function device, or a control unit. Nodes between the focal point and the entry point may perform the SNA/DS intermediate role to provide a connectionless delivery service and fan-out (that is, one copy coming into the intermediate node can be replicated and forwarded to several subsequent nodes for ultimate routing to the destinations).[9] One of the entry points may serve as the preparation site for *change files*, that is, files containing component replacements or updates and any necessary instructions to install them. This entry point, if required, is typically located at the central site with the focal point, where it can be attended. The preparation site can also be at a focal point rather than at an entry point, or even in a system that serves neither a focal point nor an entry point role (in that case, the prepared change files must be introduced by some other means, such as distribution tapes, at either a focal point or an entry point). It is the responsibility of the preparer of the change file to include in it any necessary prerequisite information to be checked by the target entry points when the change is installed. However, corequisites (a group of change files to be installed together) may be specified by the network planner.

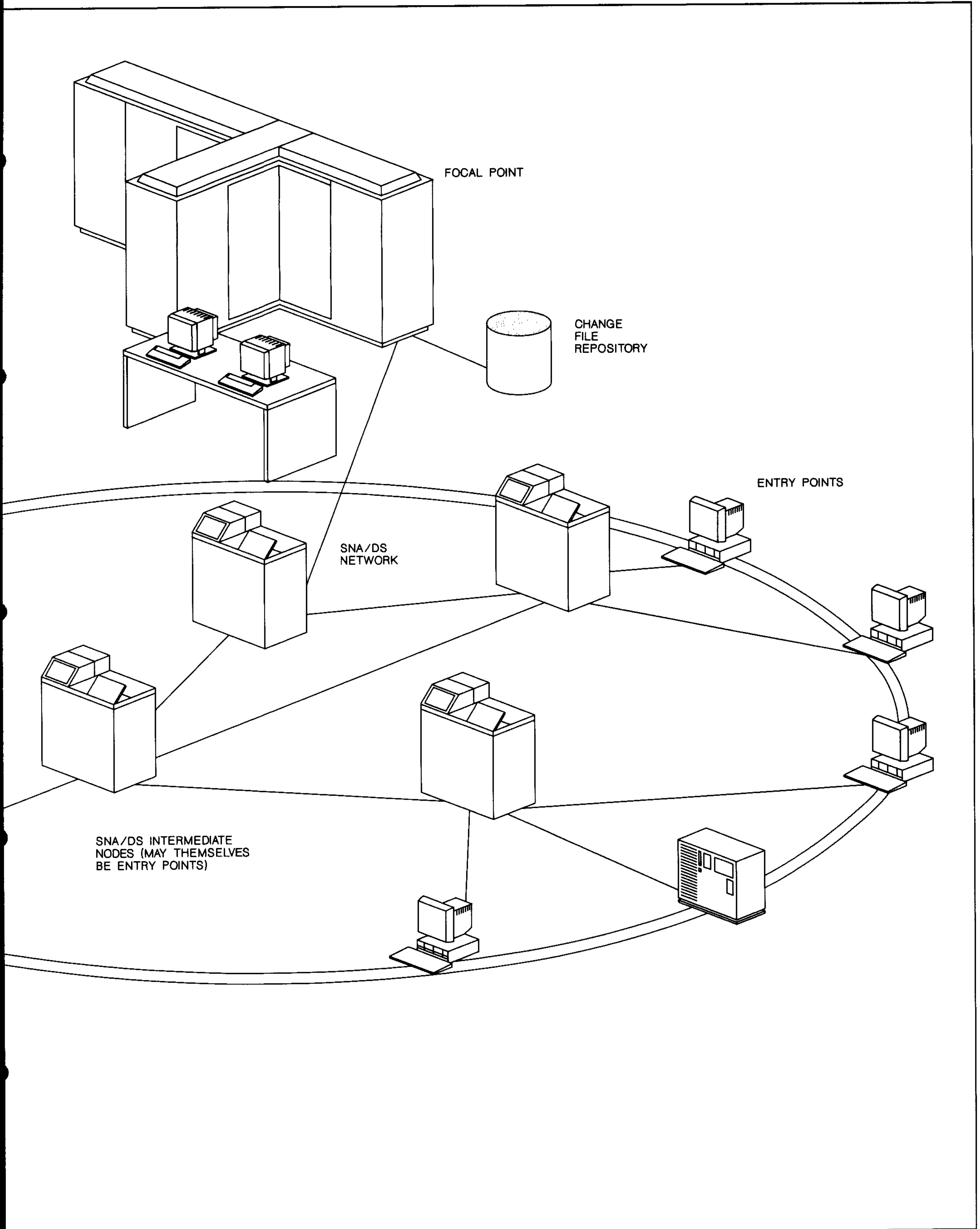
Change management requests

An SNA/MS change management focal point provides for the following requests at its interface with the network planner:

- *Retrieve* obtains a change file prepared at an entry point or at another focal point for storage at the focal point.
- *Send* distributes a change file from the focal point to one or more entry points or other focal points.
- *Delete* deletes a change file at one or more entry points.
- *Install* uses a change file and its corequisites, if any, to alter, at one or more entry points, all components necessary to effect the change. The entry point can perform such alteration in a removable manner if requested, that is, so that a subsequent request (Remove) can return all those components to their condition prior to the alteration. The network planner can request testing either before or after the installation process is performed by the entry point if the entry point supports such testing. For example, an entry point can test a new version of a configuration file for validity before deleting the old version. Also, automatic removal of changes (if the tests or installation fail) or automatic acceptance (see Accept, below) is possible. The network planner can designate components altered by the installation process for *trial* activation, or alternately, *production* activation. The designation conditions how the entry point is later reactivated.
- *Send-and-Install* is the same as Install, except that the focal point sends a change file in the same request.
- *Remove* returns all components previously altered in connection with a change to their condition prior to the installation of the change. It is possible only for changes installed previously in a removable manner.
- *Accept* relinquishes resources at an entry point required to maintain removability of a change and cancels the removability of a change installed previously in a removable manner.
- *Activate* causes reactivation of the entry point. Such reactivation uses changes installed on a trial basis as well as those installed in production. Also, the network planner can request that the entry point not attempt reactivation if user sessions are currently active at or through the entry point.

Figure 1 A network perspective




FOCAL POINT
CHANGE
FILE
REPOSITORY
ENTRY POINTS
SNA/DS
NETWORK
SNA/DS INTERMEDIATE
NODES (MAY THEMSELVES
BE ENTRY POINTS)

Testing

Without the explicit testing features provided on the Install request, the only testing would be done by an entry point user(s) over some period of time after the change was installed. If problems were encountered, the central-site network planner would need to be consulted, because only he or she would know about the previous installation of changes and be in a position to issue the Remove commands.

Two kinds of explicitly requested tests are useful *as part of the installation process*, so that the central-site network planner can be informed immediately about the results of certain diagnostic tests as part of the installation report:

1. *Pretests*—Tests made before the programming components are altered
2. *Posttests*—Tests made after the programming components are altered but before the installation report is made

Some reasons why automatic testing of changes is desirable:

- Possible corruption of the change between its initial development and installation at the entry point
- Possible sensitivity of the change to differences between the maintenance level of the target system and the maintenance level of those systems in which the change has already been tested
- Possible sensitivity of the change to differences between the configuration of the target system and the configuration of those systems in which the change has already been tested
- Possible sensitivity of the change to differences between functions or applications present at the target system and those present at systems in which the change has already been tested

The entry point performs the pretest (if requested) by examining the change file *before* components are altered. The change file contents may be examined for self-consistency and consistency with the configuration, maintenance level, or application set of the entry point. For example, an altered version of a file that contains input data to a routine can be checked to see if it contains inconsistent specifications.

If the pretest fails, no attempt is made to alter the components, and the installation report is made to the requester with the test results.

The posttest is performed (if requested and supported) by the entry point *after* components are altered but before the installation report is made. Altered versions of the components are tested directly, for example, by executing diagnostic routines. Such routines or test instructions can be distributed along with the change.

An entry point must be capable of installing a change so that it is removable.

If the posttest fails, the components are returned to their unaltered condition if another parameter, *automatic removal*, is specified. In any event, the installation report is sent with the test results, and the requester is informed immediately. If required, the requester has the opportunity to remove the change so that the impact on end users is minimized.

The activation use parameter

The *activation use* parameter of the Install request causes the entry point to install the indicated changes for *trial activation* or *production activation.* The Activate request contains a parameter that causes the entry point to activate both the trial and the production versions of altered entry point components. Entry points implement the following types of local reactivation:

1. Use of both trial and production components
2. Use of production components only

Changes that cannot be tested fully or that have a strong potential to affect the entry-point-to-focal-point communication path are best installed on trial. After activation and a period of successful operation, the changes may be installed again in production.

Of course, to provide the trial activation capability, an entry point must be capable of installing a change so that it is removable. That is, the entry point must be able to keep a copy of the production-level system. Storage capabilities at the entry points may preclude support of trial activation in some cases. If so, the installation will be refused if activation use on a trial basis is specified.

The parameters on the Install and Activate requests reflect very specific reliability requirements of entry point implementations. Although the basics of the Activate request provide the ability to reactivate an entry point after changes have been installed (for example, by loading altered microcode), there is a need to provide a way to use only unaltered versions of components during local activation. Without such a capability, reactivation of the entry point could result in the use of a change so destructive that the path to the focal point cannot be maintained.[11] Repair (in the form of further change distribution and installation) cannot be triggered by the focal point in this case. Reactivation must be triggered at the entry point through human intervention. Reactivation of the (working) production-level components can restore communication with the focal point and allow the network planner to repair the components.

Hence, the ability to store both a production and a trial set of entry point components was seen as critical to allow operator intervention to be simple, and to avoid the requirement for both change management skills and awareness of network planning activities at the entry point. Through support of a default local activation of the production system, an entry point implementation can provide hardware externals that are very simple for nontechnical users of entry points, typically not attended by technical people.

How the network planner uses NetView Distribution Manager

Focal point implementations may provide the network planner with the ability to aggregate a series of requests and specific scheduling and conditioning rules for their execution into a *change distribution plan.* In the NetView Distribution Manager implementation, plans are defined and maintained in a library, and when specified by the schedule, they are submitted for execution. The status of the submitted plan is updated as execution progresses. Control of the distribution, including recovery and restart operations, is automatically and continuously performed. The network planner builds a plan out of *phases*, each of which targets one entry point or a group of entry points. Also, files can be grouped and handled together. Each phase is built from a sequence of requests as described above. These building blocks allow the network planner to use the NetView Distribution Manager to perform the following functions:

- Start a phase at a specified date and time
- Cancel a phase not started or completed within a specified time interval
- Join phases of the same plan with conditioning rules
- Execute a procedure at the focal point when a phase ends
- Execute a plan at the same time every day, with tolerance for a specified delay

Thus, the network planner has a powerful, easy-to-use tool to manage distribution of changes in the network, or indeed the transmission of any files. The complexity of the process is reduced because NetView Distribution Manager presents these functions to the network planner in a logical sequence, either in an interactive fashion using panels or with a batch interface.

Distribution management includes the following subtasks:

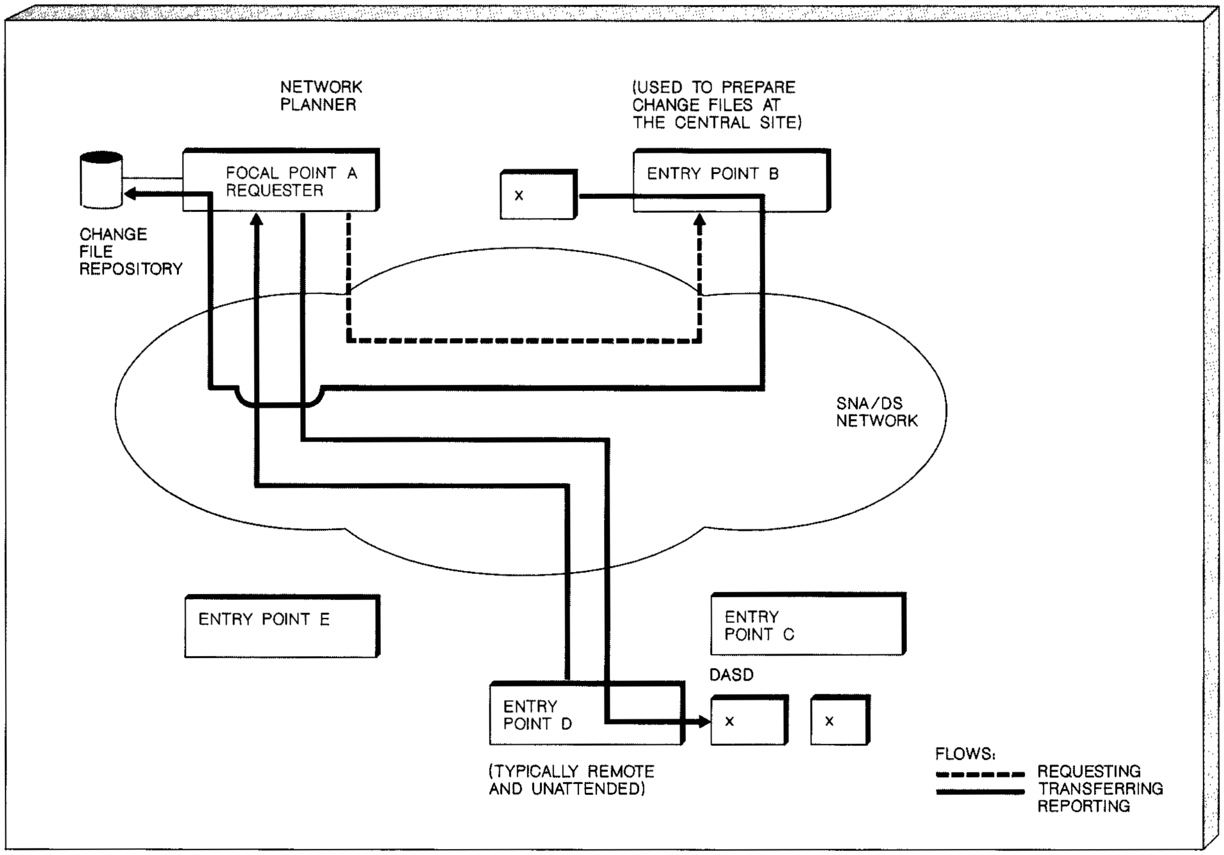
- Prepare and submit change management plans
- Maintain the files stored in the focal point repository
- Control the progress of submitted plans and evaluate the resulting reports
- Track the status of the change files by entry point

Multiple network planners may concurrently access the focal point facilities and perform similar or complementary functions. Planning activities may be concurrent with distribution activities. Plans may be validated for correctness before their submission. Status of plan execution may be tracked by the network planner. Certain recovery actions are automatically attempted if required. Distribution may be performed unattended; however, an interactive facility is available for the network planner to monitor and control the distribution operations and take appropriate corrective actions when problems are found.

Converting plan requests to commands and server instructions

Both focal points and entry points must contain SNA/DS support, including a specific SNA/FS server

Figure 2 Architectural model for change management



that is able to interpret control information. The application (or *agent*) using SNA/DS in this case is provided by SNA/MS. As can be seen in Figure 2, the focal point supports requests and replies at its interface with the network planner (that can be either a person or a program). A network planner request is converted by the focal point into a *command* that it sends to one or more entry points. For example, the Retrieve request is converted to an SNA/FS *Transfer To Requester* command. A command is executed by an entry point and results in a *report* to the focal point. Each report associated with a request is given to the requesting network planner when it is received by the focal point. Reports are defined by each of the SNA components. An SNA/DS report is received if an error occurred in the distribution network. If distribution is successful, either an SNA/FS report or an SNA/MS report is received. SNA/FS reports occur if the request was for file transfer only. SNA/MS reports indicate the success or failure of the request.

Commands and reports are carried in SNA/DS *message units*. Each message unit may contain the following items:

- A command or report, defined by either SNA/MS or SNA/FS
- SNA/FS control information
- A file

A message unit may identify but not carry a file, in which case SNA/FS control information is present without a file. An example is a Retrieve request. In contrast, a command or report need not be present. For example, a successful Retrieve request results in the return of a file and its control information but no report. These examples are illustrated in the scenario given later.

Different from both network planner requests and focal point commands is the SNA/FS *server instruction* in the control information. The server instruction indicates to the SNA/FS server how to manipulate the file into, or out of, the storage facilities. For example, a Transfer To Requester command flows with a *Fetch* server instruction, and the reply contains a *Create&Load Or Replace* server instruction.

How SNA/Management Services uses SNA/File Services global names

One of the important motivations for this architecture is the requirement to minimize the implementation cost incurred by a focal point in identifying the files used by the wide variety of products in an SNA network. SNA/FS provides a global name for a file, consisting of a set of tokens. For a description of this feature, please see Ashfield and Cybrynski.[7] SNA/FS standardizes the number of tokens allowed (up to 10), the size of each token (up to 16 characters), total name length (65-n where n is the number of tokens), and the character set allowed for the token values (a limited set of character graphics displayable on most types of displays). In addition, SNA/FS architecture maintains a registration of values for the highest-order token. For example, MCODE is the registered value for change files containing microcode. SNA/MS maintains a registration of the values of some of the other tokens, delegating authority in some cases to administrative organizations. For example, the IBM machine type is the second token for microcode. As a result, a focal point is required to implement only one input panel, say, for a user to identify microcode files for a potentially wide variety of types of target entry points.

Definitions of token values by SNA/MS were made to satisfy two general requirements: First, each file containing a change must be uniquely identified; second, the user must be provided with some idea of the type and identity of the change when displaying the name, or an application must be allowed to process the token values. For example, the file name MCODE.9135.NA.PATCH.1234 indicates that the file contains a patch rather than an engineering change. This is needed to uniquely identify the file and also to provide useful information on display.

It is advantageous for the user to create (or have provided by product developers) change files that can be installed on a large number of entry points. Such change files allow the fan-out feature of SNA/DS to be fully exploited and reduce the user's effort. For example, files containing microcode can be designed to be applicable to many control units, whereas those containing customizing data are specific to individual control units.

The types of change files applicable to SNA/MS are microcode, customizing data, software, procedures, applications data, and documentation. The first product offering and architecture release support

SNA/MS makes use of SNA/FS partial name processing for both retrieval and distribution.

microcode and customizing data. Within each of these types, SNA/MS defines some of the lower-order tokens, and some additional token values are defined by product implementations.

SNA/MS makes use of SNA/FS *partial name processing* for both retrieval and distribution. Partial name processing is used when the network planner wishes to specify only some of the file identification tokens (typically, the higher-order ones). An example of this is when the latest version of a customizing data file is to be retrieved, but the user cannot remember the version number contained in the lowest identification token (and does not care what its value is).

On distribution, partial name processing may be needed to specify which change file to destroy to make room for a new one when entry point storage constraints arise. An important requirement addressed by the architecture is that destruction take place only when installation has been requested properly and pretests have been performed successfully. The complete identity of the file replaced is included in the report of successful installation.

A scenario

The following scenario illustrates how the network planner uses the change management functions provided by SNA/MS.

Consider a network planner working at a focal point A, whose job is to prepare changes at a local entry

Figure 3 The network planner submits a plan to the focal point

```
At 14:00 EDT on 7/9/89, do the following:

  Retrieve change file 'x' from B  *
  Retrieve change file 'y' from B  *
  Send 'x' to D  *
  Send And Install 'y' with 'x' as corequisite on trial at D  *
  Activate D using trial and production changes

At 14:00 EDT on 7/23/89, do the following:

  Retrieve change file 'w' from B  *
  Retrieve change file 'z' from B  *
  Send 'x' to C and E  *
  Send And Install 'w' with 'x' as corequisite in production at C  *
  Send And Install 'z' with 'x' as corequisite in production at E  *
  Activate C and E using trial and production changes

*If this step is successful, then perform the request following
```

point B and distribute them to a number of remote entry points—C, D, and E. A new microcode change 'x' is received from IBM and prepared at B for network installation. Since a corresponding change to customizing data is required to use a new function introduced by 'x', the network planner also prepares three change files, 'w', 'y', and 'z', containing the customizing data for entry points C, D, and E, respectively. This preparation is done at entry point B, where the files are created and stored. The network planner prepares and submits to the focal point the change distribution plan shown in Figure 3.

The following discussion shows how the plan is accomplished by the focal point according to the architecture. The accompanying figures illustrate the process.

First, the Retrieve request is converted to an SNA/FS Transfer To Requester command. The SNA/FS server at the focal point encodes a server object containing the file name, and the SNA/FS server at the entry point decodes it (Figure 4).

The SNA/MS agent at entry point B instructs its SNA/FS server (in the *source* role) to fetch the file and its SNA/DS component to send it to the focal point. The file is sent without a report, and the SNA/MS agent at the focal point is signaled by its SNA/FS server that the file has arrived (Figure 5). The SNA/FS server at the focal point (in the *target* role) stores the file into the repository, and the next request in the plan can proceed. For a description of the SNA/FS server roles, please refer to Ashfield and Cybrynski.[7]

The retrieval of the change file 'y' containing the customizing data is performed successfully in the same manner.

The next request is to send 'x' to D. The SNA/MS agent at the focal point instructs its SNA/FS server (in the source role) to fetch 'x' from the file repository and builds and sends an SNA/FS Report FS Action command that is carried with the file (Figure 6).

The SNA/FS server at D (in the target role) stores the file, and the entry point SNA/MS agent obeys the command by building and sending an SNA/FS report to the focal point (Reporting FS Action) (Figure 7).

Next, the customizing data are sent, and installation of both 'x' and 'y' as corequisites is requested in the same flow. In this case (Figure 8), the file is sent with an SNA/MS Install command (instead of an SNA/FS Report FS Action command). The SNA/FS server at the entry point stores the file, and the SNA/MS agent installs 'x' and 'y'.

An SNA/MS report (Reporting Installation) is built and returned to indicate successful installation (Figure 9).

Figure 4 The focal point retrieves a change file

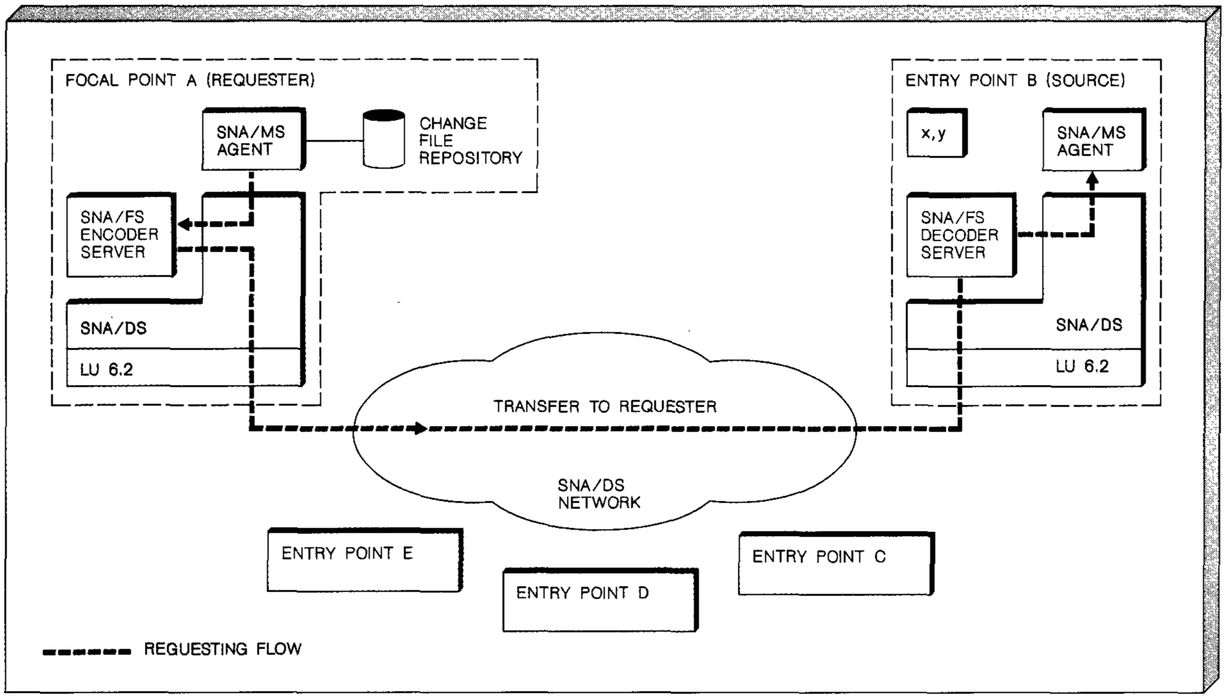


Figure 5 The change file is returned

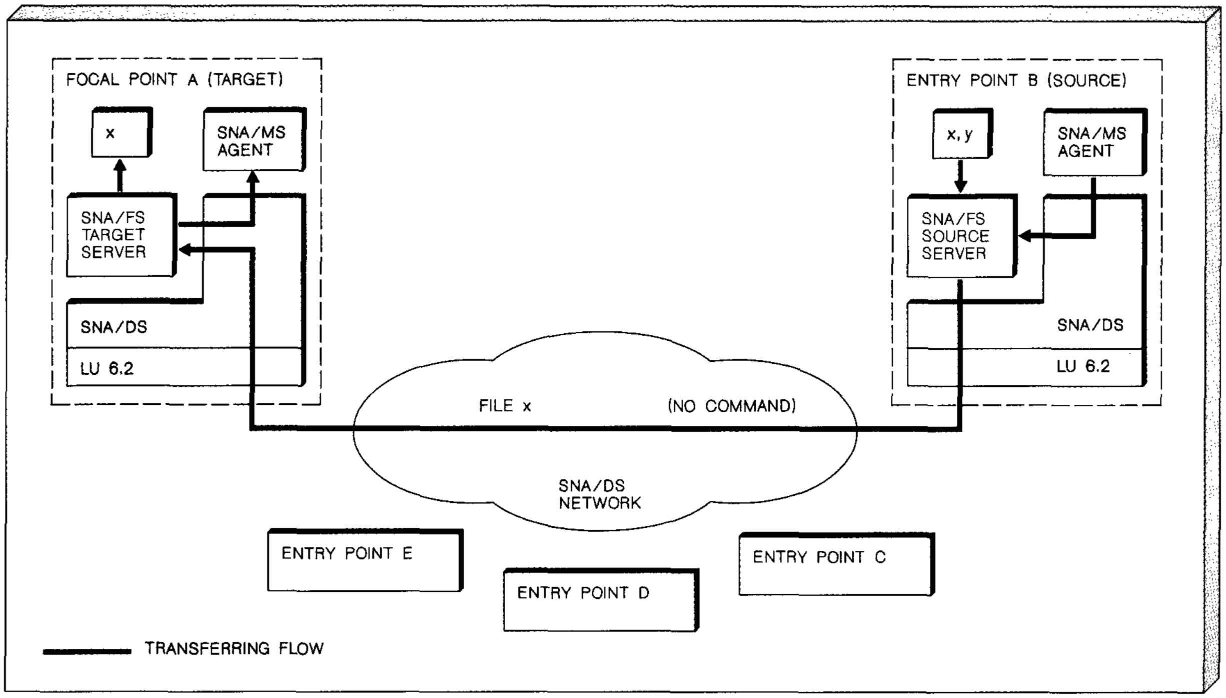


Figure 6 The focal point sends a change file

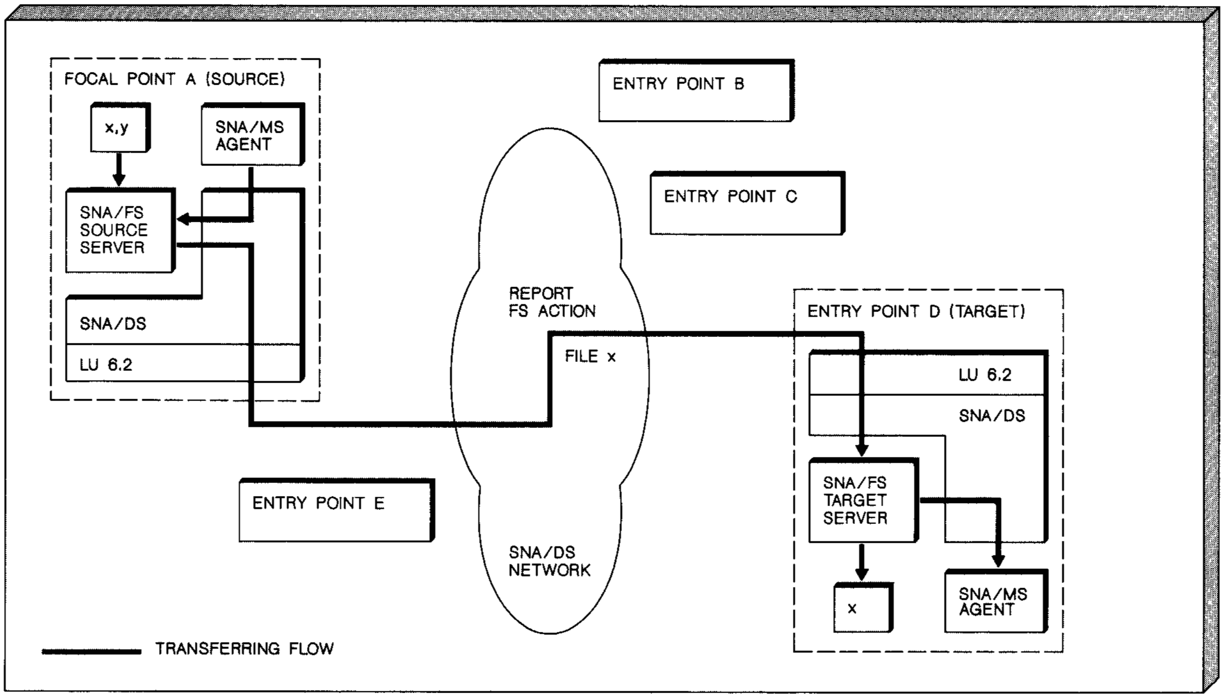


Figure 7 The entry point reports successful file transfer

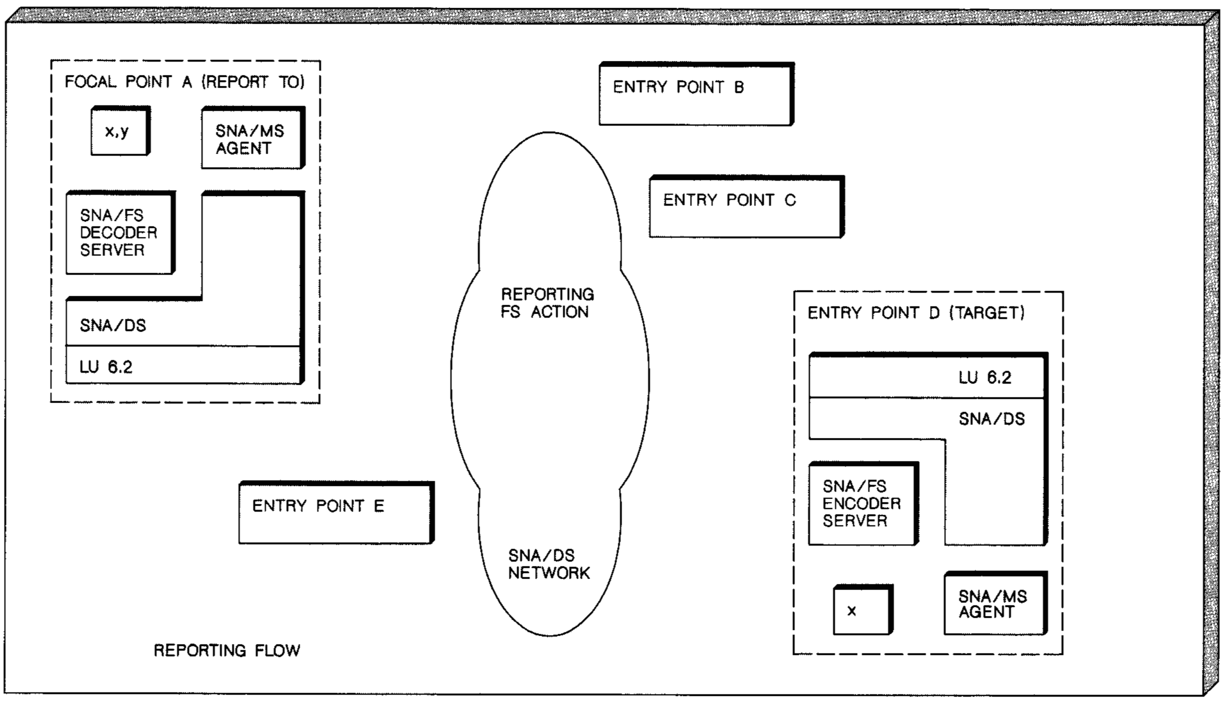


Figure 8 The focal point sends and installs a change file

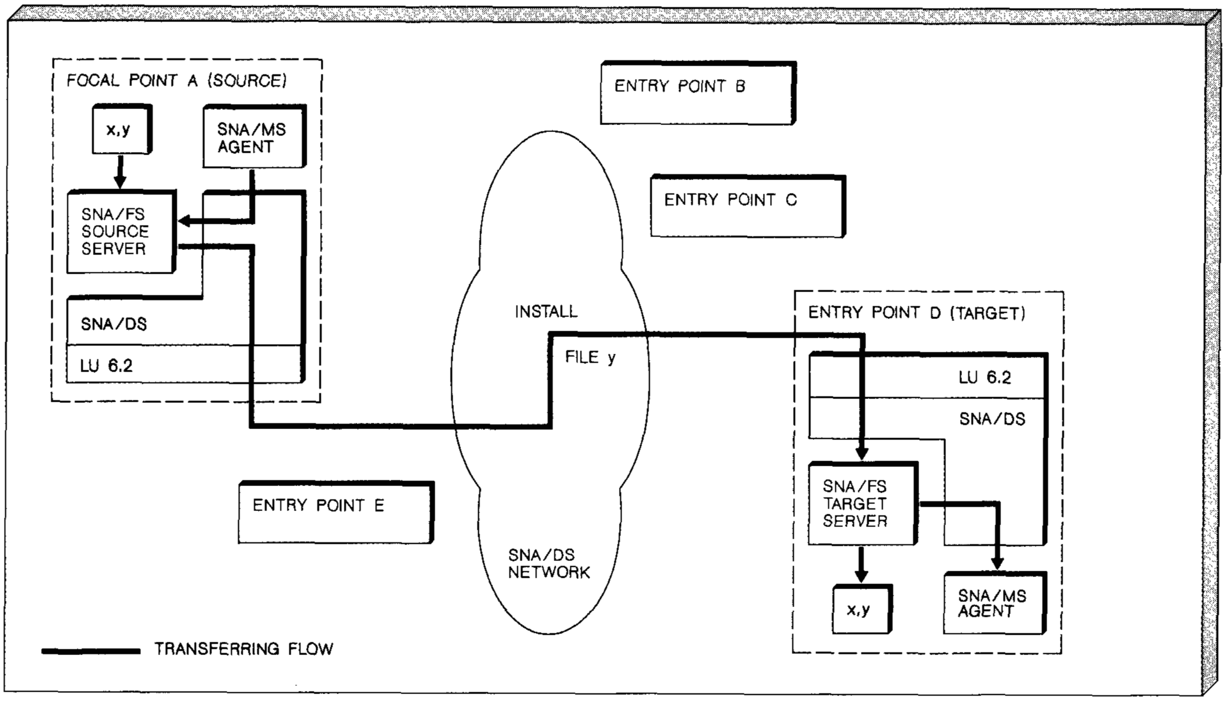


Figure 9 *The entry point reports successful installation*

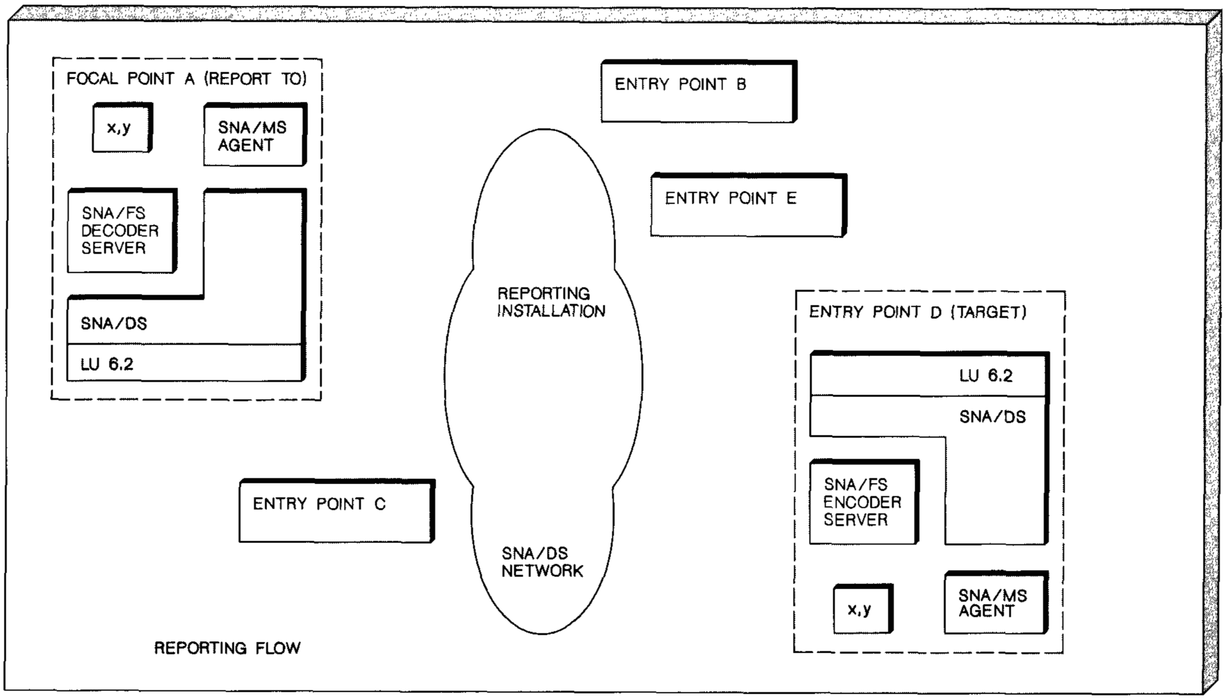
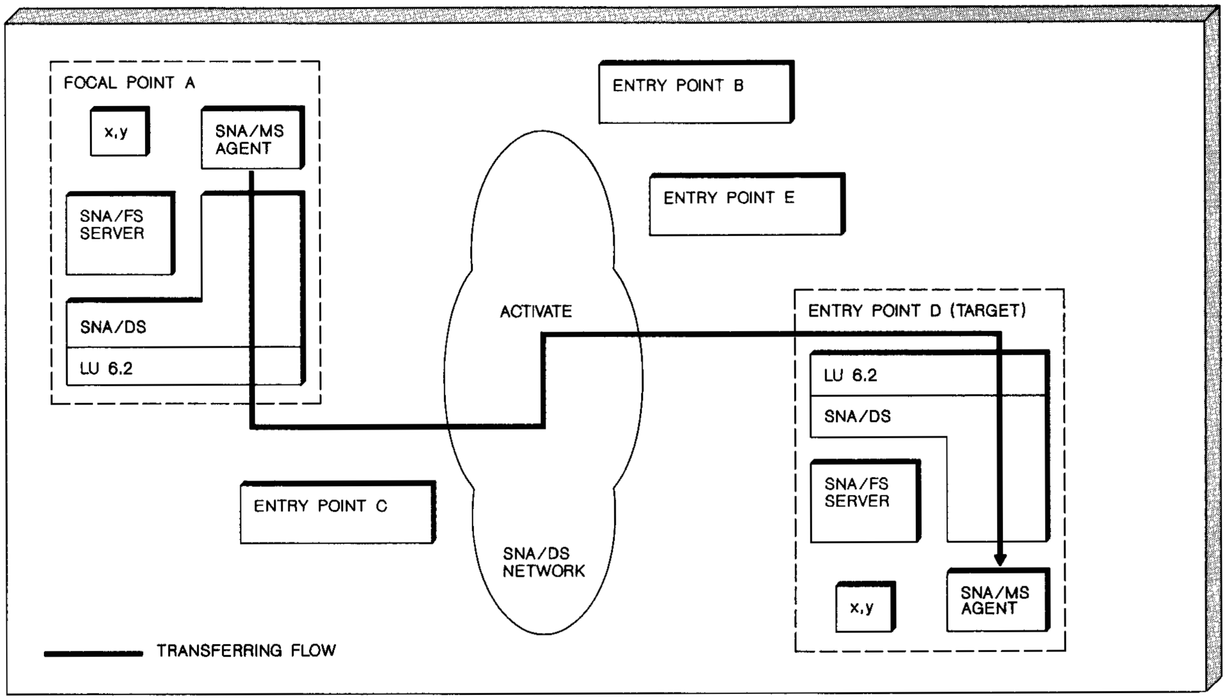


Figure 10 The focal point reactivates an entry point



Reactivation of entry point D is requested, and the command is sent to D (Figure 10). The SNA/FS servers at the focal point and entry point are not involved.

The entry point reports that reactivation will be attempted, and the plan is complete.

Reactivation of the entry point includes termination of the session between A and D since the entry point completely reinitializes itself.

Two weeks later, at the time and date specified in the plan, the customizing data for the two remaining entry points are retrieved and the new microcode is sent and installed at C and E as well, this time in production. If the new microcode had caused problems at D, the network planner could have canceled the plan to prevent this final step.

Summary

SNA/Management Services has been enhanced to provide change management capabilities. A manager responsible for operation of an SNA network can use it to plan, schedule, and track changes to SNA nodes that are typically remote and unattended, in a nondisruptive fashion, during their normal operation. Application developers can use it to distribute and install new software and associated application data throughout a network where an application is distributed.

Acknowledgments

The authors gratefully acknowledge the help of the many designers, developers, and architects involved in this project. Special thanks go to Arthur A. Daniel of AS/400 Systems Design for many of the initial ideas, James C. Ashfield, Ercole Colonese, Donna B. Cybrynski, Armando Ferrauto, Florian K. Kandefer, George M. McMullen, Joseph K. Parks, and Frank J. Petsche for significant design and development contributions, and Pasquale Autru, James P. Frey, and Matthew L. Hess for welcome management guidance and support.

Cited references and note

1. R. J. Sundstrom and G. D. Schultz, "SNA's first six years: 1974–1980," *Fifth International Conference on Computer Communication*, Atlanta, GA, North-Holland Publishing Co., Amsterdam (September 1980), pp. 578–585.
2. R. J. Sundstrom, J. B. Staton III, G. D. Schultz, M. L. Hess, G. A Deaton, Jr., L. J. Cole, and R. M. Amy, "SNA: Current requirements and direction," *IBM Systems Journal* **26**, No. 1, 13-36 (1987).
3. D. B. Rose and J. E. Munn, "SNA network management directions," *IBM Systems Journal* **27**, No. 1, 3–14 (1988).
4. R. E. Moore, "Utilizing the SNA Alert in the management of multivendor networks," *IBM Systems Journal* **27**, No. 1, 15–31 (1988).
5. *SNA/Management Services Reference*, SC30-3346, IBM Corporation (1989); available through IBM branch offices.
6. *NetView Distribution Manager General Information Manual*, GH19-6587, IBM Corporation (1989); available through IBM branch offices.
7. J. C. Ashfield and D. B. Cybrynski, "System-independent file management and distribution services," *IBM Systems Journal* **28**, No. 2, 241–259 (1989, this issue).
8. *SNA/File Services Reference*, SC31-6807, IBM Corporation (1989); available through IBM branch offices.
9. B. C. Housel and C. J. Scopinich, "SNA Distribution Services," *IBM Systems Journal* **22**, No. 4, 319–343 (1983).
10. *SNA/Distribution Services Reference*, SC30-3098, IBM Corporation (1989); available through IBM branch offices.
11. Testing is optional at installation time, and implementations of testing cannot always verify the change. Software or microcode defects generally cannot be detected during an Install test function.

Christopher P. Ballard *IBM Italy Telecommunications Development Center, Viale Oceano Pacifico 71/73, 00144, Rome, Italy.* Mr. Ballard is a staff programmer designing network management products. He joined IBM in 1977 in the Field Engineering Division in San Francisco, California, where he specialized in telecommunications program support for large System/370 environments. In 1983, he transferred to SNA/Management Services development, working first on network management of link connections, then leading architecture development for change management and remote operations. He was assigned to the Rome center in 1989. Mr. Ballard holds a B.Sc. (hon.) in mathematics from University College, London, and an M.B.A. in telecommunications management from Golden Gate University, where he taught data communications in the graduate program.

Livio Farfara *IBM Italy Telecommunications Development Center, Viale Oceano Pacifico 71/73, 00144, Rome, Italy.* Mr. Farfara joined IBM in 1970 and was the development manager responsible for the implementation of NetView Distribution Manager Release 2. He is currently the Rome TDC design manager for Communication Systems products. In 1979 he joined the IBM Telecommunications Development Center and participated in the design and development of the Distributed Systems Executive product. He also contributed to the design of SNA/File Services and SNA/Management Services change management. Mr. Farfara earned a Laurea degree in mechanical engineering from the University of Pisa in 1970.

Barbara J. Heldke *IBM Data Systems Division, P.O. Box 100, Kingston, New York 12401.* Ms. Heldke joined IBM in 1974 and is currently a senior programmer in the software manufacturing organization. In an earlier assignment within the former Communication Products Division, she was the designer for the IBM 3174 change management implementation, a role that involved significant participation in the development of SNA/File Services and SNA/Management Services change management. Previous experience included product design for SNA extensions to ACF/VTAM and NCCF. Ms. Heldke holds a B.S. in mathematics and computer science from the University of Wisconsin.

Reprint Order No. G321-5358.