Burroughs

Burroughs Network Architecture (BNA)

ARCHITECTURAL DESCRIPTION

REFERENCE MANUAL VOLUME 1

PRICED ITEM

Printed in U.S.A. April 1981 1132172

Burroughs

Burroughs Network Architecture (BNA)

ARCHITECTURAL DESCRIPTION

REFERENCE MANUAL VOLUME 1

PRICED ITEM

Printed in U.S.A. April 1981 1132172

"The names used in this publication are not of individuals living or otherwise. Any similarity or likeness of the names used in this publication with the names of any individuals, living or otherwise, is purely coincidental and not intentional."

Correspondence regarding this document should be addressed directly to Burroughs Corporation, Box CB7, Malvern, PA. 19355, Attn: Systems Documentation Dept., TIO East.

LIST OF EFFECTIVE PAGES

Page	Issue
Title	Original
ii, iii	Original
iv	Blank
v thru x	Original
1-1 thru 1-3	Original
1-4	Blank
2-1 thru 2-36	Original
3-1 thru 3-17	Original
3-18	Blank
4-1 thru 4-14	Original
5-1 thru 5-92	Original
6-1 thru 6-5	Original
6-6	Blank
7-1 thru 7-14	Original
8-1 thru 8-6	Original
9-1 thru 9-51	Original
9-52	Blank
1 thru 6	Original

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

INTRODUCTION

Burroughs Network Architecture (BNA) is a precise architectural plan for using Burroughs standard products as building blocks for networks. Under BNA, networks provide users with an environment in which they can develop and operate distributed application systems.

In BNA, complex distributed processing functions are accomplished by a sophisticated architecture. An architectural description is employed in this manual to facilitate an understanding of certain portions of BNA. However, it is fundamental to BNA that its use appears to the user to be a simple extension of standard non-network operations. The descriptions of user interface in this manual and others will illustrate that ease of use; in the user interface for Inter-Process Communication, the user interface for Host Services, and in Network Control.

BNA defines the functions, protocols, formats, and general structure of the unified communication design. The design is implemented in a variety of Burroughs standard products. This manual, and the companion BNA Reference Manual, Volume 2 (Network Control), form number 1132180, describe the architecture but do not describe any individual product implementation of the architecture. They are intended to provide the reader with an overall understanding of the architecture, which is essential to an understanding of the implementations. For specific details on any individual Burroughs product implementation of BNA, refer to the publications for that product.

An overall understanding of the architecture of BNA is needed to plan BNA networks, and to establish network-wide basic usage, operation and recovery procedures. Additional system-dependent BNA information is necessary for final planning and operation and is supplied in individual system documentation.

SECTION 1

HOW TO USE THIS MANUAL

This manual presents an architectural description of Burroughs Network Architecture (BNA). The manual is organized to accommodate readers with four levels of interest:

1. Operators

2. Applications Programmers

3. Network Planners and Operations Supervisors

4. Network Systems Programmers.

Figure 1-1 shows the recommended reading and reference sequences for each of these interest levels. It also indicates the expected depth of understanding required by each of these interest levels from each of the nine sections of this manual.

The manual is intended to be used in conjunction with the BNA Reference Manual, Volume 2 (Network Control), form number 1132180, and with the BNA product implementation documentation for the specific Burroughs systems with which the user will be involved.

A brief description of each of the sections in this manual is given in the following paragraphs.

Section 2, OVERVIEW, provides a general overview of Burroughs Network Architecture (BNA) and a description of the features, functions, concepts, and scope of BNA. The reader should have a basic understanding of data communications. This section is intended as a stand-alone overview and also as an introduction to the other sections. All readers should start with this section.

Section 3, INTER-PROCESS COMMUNICATION, Functional Description, provides a greater level of detail in the use of BNA to effect communications between user processes within the network. This section describes the user interface and is primarily intended for the Application Programmer, but should be understood by Network Operation Supervisors and Network Systems Programmers.

Section 4, HOST SERVICES, Functional Description, provides additional detail on the Host Services features and capabilities, in order that the use of these may be properly planned. This section describes the Host Services user interface and is primarily intended for the Network Operations Supervisors and Network Systems Programmers and Applications Programmers.

Section 5, FUNCTIONAL DESCRIPTION - NETWORK SERVICES, provides a greater level of detail and understanding for persons such as Network Planners, Network Systems Programmers or Network Operations Supervisors who will work directly with a network of Burroughs systems using BNA. The information in this section establishes the background needed to configure/reconfigure nodes or links, to plan and execute recovery procedures, develop interfaces with existing networks, etc.

Section 6, INITIALIZATION, is intended for Network Operations personnel and Network Systems Programmers. It describes the process of configuration, reconfiguration, and initialization of the network. It does not provide specific details because these are unique to a particular system and are described in the separate BNA Product System documentation.

Section 7, ACCESS CONTROL, is intended for Managers, Operations personnel, Programmers, and others with a responsibility for security. It describes the features provided to protect the network from accidental or malicious intrusion, including verification, authentication, greetings, and identification.

Section 8, LOGGING AND MONITORING, is also intended for Network Operations personnel. It describes the features which can be used to monitor network operation. The data gathered by the monitoring functions can be used for billing purposes; in the matching of existing or available network resources to the workload; and in the minimizing of the effect of malfunctions in nodes or links.

Section 9, FRAME FORMATS, provides a detailed breakdown of the message formats at each of the levels of Network Services and at the Host Services level, and the relationships between them. This information enables a user to interpret the output of a line monitoring device connected to the communication links. This section also provides very detailed reference material to be used along with the other sections.

The separate Burroughs Network Architecture Reference Manual, Volume 2 (Network Control) is a reference document primarily intended for the experienced Network Systems Programmer, Planner or Operations Supervisor. It provides detailed and extensive data on the syntax, semantics, restrictions, and use of Operations Interface commands, responses, reports, etc.

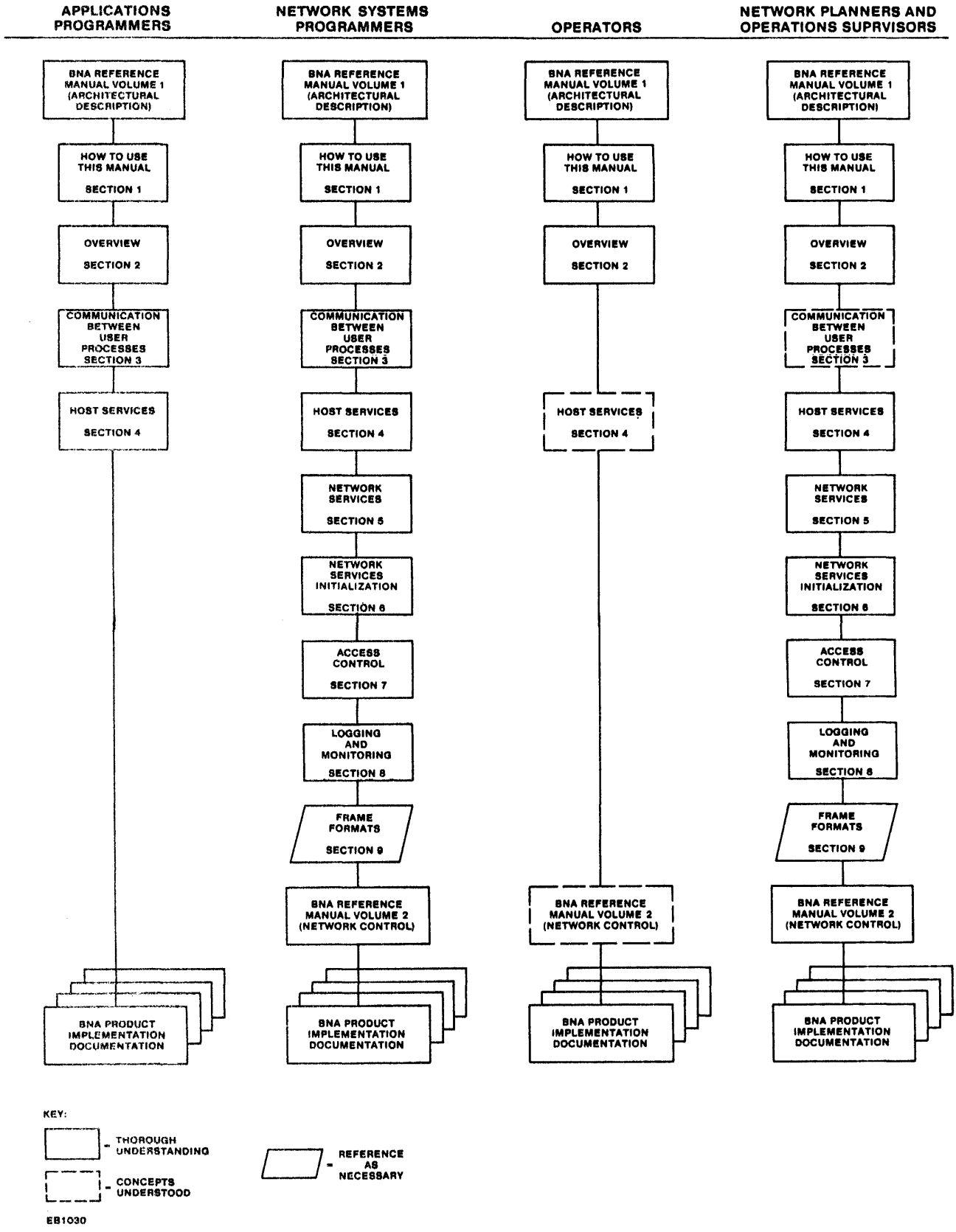


Figure 1-1. Recommended Reading Sequences

SECTION 2
OVERVIEW

INTRODUCTION

This section provides an overview of Burroughs Network Architecture (BNA) in terms of its functions, capabilities, and general organization. Introductory paragraphs describe the general trends in computer usage, Burroughs philosophy of computer usage, and the evolution of BNA as a natural extension of these trends.

GENERAL TRENDS IN COMPUTER USAGE

Computer usage from the 1960s into the 1980s has evolved through three basic phases:

1. Decentralized Processing (1960s)
2. Centralized Processing (1970s)
3. Distributed Processing (1980s)

This evolution was heavily influenced by advances in two major areas of information processing, namely, the decreasing costs of computers, and improvements in communications facilities. These are discussed in the following paragraphs.

Decentralized Processing (1960s)

With this approach, individual departments of an organization, or perhaps geographically dispersed parts of an organization, each had their own individual computer which provided the information processing for that part of the organization. At this time, communication facilities were not particularly powerful, and the Decentralized computers operated primarily in the batch environment, although each could support a small number of locally connected terminals.

One of the main characteristics of this approach was good end-user and local management involvement in the applicational use of the systems. As each department or part of the organization had its own computer, the end-user of that computer felt involved with the application of the system.

However, there were also certain disadvantages with this Decentralized approach to information processing.

The major disadvantage was that the data, the information, and the files which were associated with each computer were scattered throughout the organization.

The files were locally connected on each individual computer and the computers were not connected to each other by communication facilities. Therefore, information as a whole was not available throughout the whole organization, although there was some limited manual exchange of information across systems.

A second problem was that there was a lack of coordination of overall planning for application design and operation, because each individual operational unit of an organization implemented its own computing facilities.

Also at this time, during the 60's, the cost of a computer was very high, and therefore it was very difficult for an organization to buy and install multiple computers in different departments.

Centralized Processing (1970s)

The trend during the 1970's, as communications facilities improved, was toward Centralized processing.

The major characteristic of this approach was that one computer site provided all the information processing for the organization, and that all information, all data and files, was gathered together into that one site.

Each individual department, and user of the computer applications, was connected by terminals via communications links to that computer site.

Therefore information could be made available throughout the whole organization. However, there were also some disadvantages with this Centralized approach.

First, there was a lack of end-user involvement because now the computer had been taken out of the hands of the end-users and local management, and was normally controlled by a separate Data Processing department. In many cases this made it more difficult to maintain the same end-user and local management involvement with the application of the computer.

Another disadvantage, in those cases where the organization was geographically dispersed, was that the costs involved in connecting many terminals to the one computer site became high. Therefore, the overall communications costs represented a significant part of the total cost of the computer.

There was also, in the Centralized approach, an availability risk. For example, if there was a problem with that one site, then all information processing for the whole organization was could be severely disrupted.

Distributed Processing

The trend toward the latter part of the 1970s, and through the 1980s, is seen to be into the world of Distributed Processing. This combines the best characteristics of both Centralized and Decentralized processing.

With this approach, multiple computers are again installed in different parts of the organization, but now they are connected by communication facilities. Although information and files may be physically located separately with the

individual computers, logically they are still available throughout the whole organization.

By returning local processing power to individual units of an organization, the end-user involvement in the application of the computers should improve. Also, because a lot of the processing can be handled locally by the distributed computers, the overall communications costs should decrease, compared with the costs involved in linking terminals to one central site.

The overall availability of information processing for the organization should improve, and availability should be improved, because no single failure of any one computer should disrupt totally the information processing capability of the organization. This trend into Distributed Processing is really only practical because of two main reasons. The first is that the decreasing costs of computers now make it economically feasible for an organization to buy multiple systems. The second is the overall improvement in communications technology, both the hardware and the software capabilities to support networks of computers.

It is likely in most cases that Distributed Processing with a range of capability of information processing scattered throughout an organization, will in fact better match the real computing needs of the organization.

BURROUGHS PHILOSOPHY OF COMPUTER USAGE

Burroughs philosophy through each of these phases has been to make the use of computing power as easy as possible at the application level, thereby minimizing the time, resources, and capital required to develop applications.

Burroughs provides Environmental Software facilities which surround and support the application programs, and insulates them completely from such physical problems as data base organization, communications terminal network management, etc. Physically dependent aspects, such as where exactly in the disk pack a particular record resides, or the physical management of getting information from one terminal in a network to a computer, is not a concern of the application programmer, but is part of the general facilities offered by Burroughs.

The goal of Environmental Software is to allow the application designer, and programmer, to concentrate exclusively on only those parts of the complete system which are specific to the application, and to ignore any physically dependent aspects such as the location of a record on a disk pack.

Essentially all Burroughs products from the very small systems to the very largest have a similar structure to support communications networks. There are two major elements in this Environmental Software structure. First a Network Control Program and second, a Message Control System. These are described in the following paragraphs.

Network Control Program

The Network Control Program performs all the functions that are directly related to physically controlling a data communications network.

The Network Control Program is written in the high level language called Network Definition Language, or NDL.

NDL is both a descriptive language and a programming language, and it allows a simple description to be made of the network configuration. For example, how many lines, which terminals are connected on which lines, etc.

In addition, the various line protocols required to control information flow on the different lines can be specified.

NDL is thus used to specify all the physical aspects of the communications network.

The application programs know nothing about any of these physical aspects. Therefore the terminal configuration can be changed, new terminals can be added, new lines can be added, or terminals can be moved, and no change will be required in any of the application programs, because there is no code, or no involvement of the application programs, in any of these physical features of a communications network.

Message Control System

The second major component of the communications Environmental Software, is the Message Control System.

The MCS serves together with the Network Control Program to make application programs transparent to both physical network topology and management.

For example, the Message Control System performs "message routing" between terminals and applications programs. This can be dynamic, based upon some transaction code in the text of the message received from the terminal. For example, a terminal operator could enter a message at any time, with a certain transaction code in it. When the Message Control System receives the message it examines the transaction code, and routes the transaction to the required program.

The very next message from the same terminal could contain a different transaction code, which would result in this message being passed to a completely different application program.

This "transaction routing function" of the MCS removes any "physical link" between any particular terminal and a particular application program.

Another function provided by the Message Control System is that of "message reformatting". For example, let us assume that we have currently a network with terminals of 2000 character screen capacity, and that an application program produces a response which will fill that 2000 character screen. If a new type

of terminal was connected to the network with a screen which only has 1000 characters, the Message Control System can take the response, and break the 2000 character image into 2 separate screens, each of 1000 characters which will be displayed, first as one page, and then as a second page, upon the new type of terminal. This can be done dynamically depending upon which terminal in the network was the destination of the message.

MCS can enforce "sign-on" procedures in order to control which terminal operator is allowed to use which terminal, and also to control which terminal is allowed to use which computer program within the computing system. Also the MCS can ensure that only authorized transactions are performed by individual terminal operators. The Network Control and Reconfiguration facility in MCS permits the system operator to receive reports, and make inquiries, related to network management, and performance statistics, etc., and also to dynamically reconfigure the network.

The interface between application programs and the Message Control System is very straight-forward, and essentially, application programs (and application designers and programmers) do not have to use any special techniques related to a communications network. Application programs read and write messages to and from the Message Control System in the same way as if the application program was reading and writing records to a file on a disk-pack.

BNA AS AN EXTENSION OF ENVIRONMENTAL SOFTWARE

The goal of BNA is to provide the environmental software in a distributed processing environment which allows the user to access the distributed resources in a network in the same manner as he presently accesses local resources.

Whereas NDL and MCS allow the applications programmer to be unconcerned with physical aspects of the communications network in a centralized processing environment, BNA will perform these functions in a distributed processing network.

Some of the tasks which have to be accomplished within Distributed Processing Networks are, for example, to allow a program or an operator to access remote files and data bases, which are physically located on some other system in the network. Also, it is generally necessary to transfer files between systems. Further, just as programs now, within one system, communicate with each other and send messages and talk to each other, so in a Distributed Processing environment, it should be equally straight-forward for programs to communicate with each other across a network.

And finally, a very important aspect is that the total resources available throughout the network should be made available as a common pool. The users should be able to share the resources available throughout the network. These resources can be the processors themselves (processing power), peripherals, terminals, data bases, and the information which is scattered around the network. All of these resources should be accessible and available as a whole to the organization, subject of course to any required security restrictions.

One very important aspect of the "availability of resources" is that of "flexibility of resource location". It is always very difficult for an organization to know exactly where is the best place to put the processing power, information, and the terminals.

Wherever we anticipate certain resources as being best located, it is very likely that at some future date an alternative location would be better suited for the placement of these resources. Therefore it is very important that we be able to place the resources at any location, and even more important that we be able to relocate some, or all, of these resources at any time without having to change the application program. A simple run-time file equation of "HOSTNAME" is the only change required when remote resources are relocated.

In summary, BNA provides the necessary Environmental Software to support applications in a distributed environment, and to allow users to gradually evolve into the world of Distributed Processing. NDL and MCS, the products which were available for communications prior to BNA, continue to exist, and continue to be used to support local networks of terminals and terminal controllers. They continue to remove the physical complexities of management of such terminal networks from the applications programs. BNA is another layer of Environmental Software, which makes available the resources of a distributed network to the application programs, without making the application programs conscious of the physical complexities of reaching and using these distributed resources.

BNA is really the continuation, the next stage in the evolution of Burroughs basic philosophy of making computing power as easy-to-use as possible at the application level.

BNA will allow users to evolve straight-forwardly, and easily, and gradually, into the world of Distributed Processing.

GENERAL DESCRIPTION OF A BNA NETWORK

Burroughs Network Architecture allows the interconnection of multiple systems from different families of the Burroughs product line. Figure 2-1 shows a typical BNA network. Each system in this figure is identified by a host name (A,B,C,D), and may be a different system type and/or configuration.

Although each host supports its own users, it is also interconnected in a BNA network to the other hosts in the network.

Each connection point in the network is called a node, and is identified by a node address (e.g. node 1,2,3,4). Two nodes which have direct connections (e.g. nodes 1 and 2) are called neighbor nodes. Communications between non-neighbor nodes (e.g. nodes 1 and 3) is accomplished by routing from neighbor to neighbor. For example, node 1 communicates with node 3 via its neighbors (node 2 or 4). Routing tables at each node keep track of the most efficient route to any other node in the network. These tables are dynamically changed as nodes or links are added to or deleted from the network, or when the physical characteristics of any node or link are changed.

Each host system in the network is considered a cooperating peer. Normally host resources are available at each node. These may be made selectively available to other hosts at the discretion of the local system, or may be denied, so that the node acts only as a routing mechanism within the network. Each node declares its degree of participation in the network via locally programmed attributes which describes its capabilities and constraints on the use of its resources. It may selectively allow or deny use of its resources to individual hosts, stations or users.

At each node in a BNA network, regardless of system type, two functional levels are defined. These are called Host Services and Network Services.

Host Services provides users and programs with the "system" functions that are needed to operate in a Distributed Processing environment.

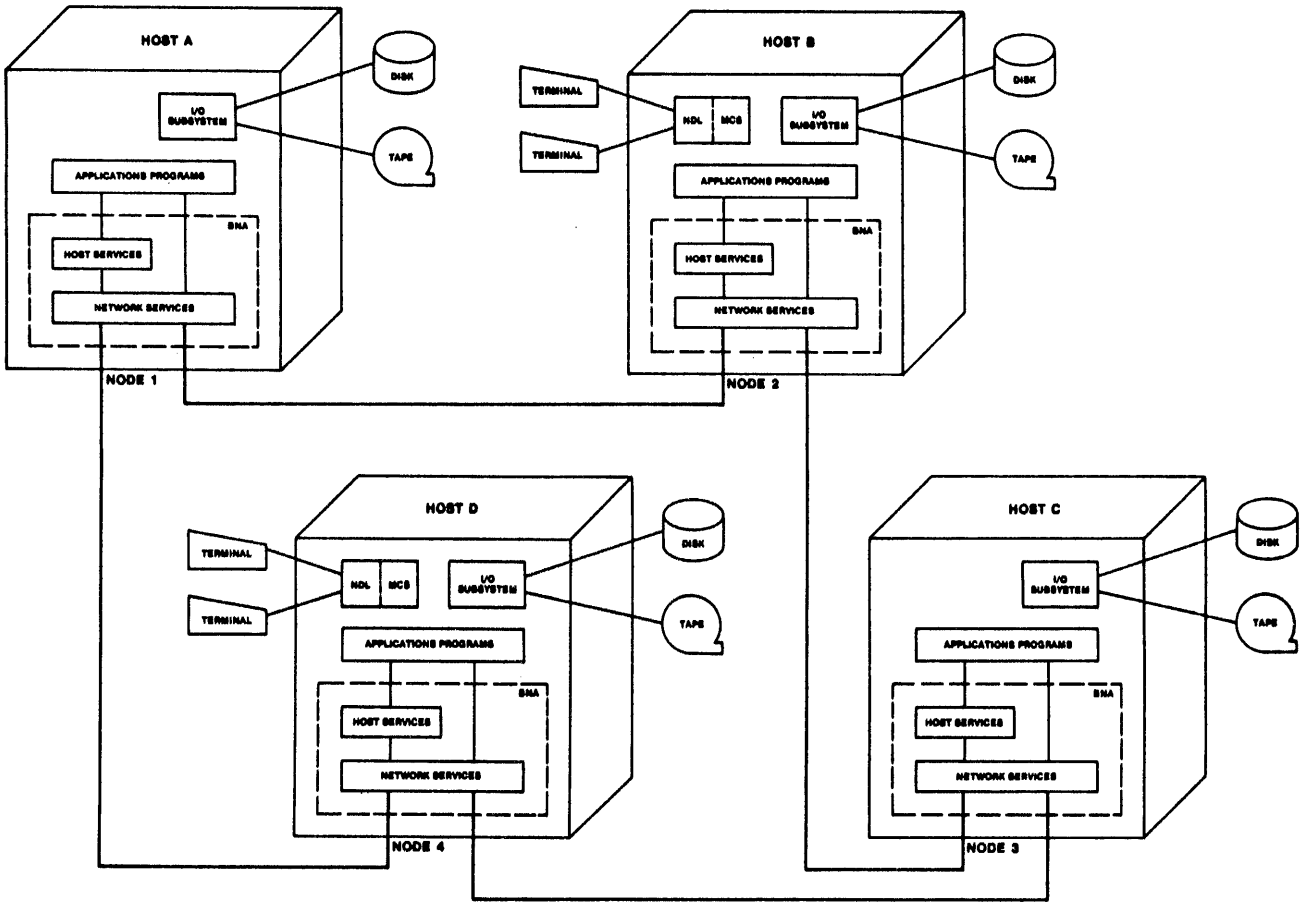


Figure 2-1. Typical BNA Network

Host Services can be viewed as the extension to the functions provided by an operating system in one single computer site. The extensions allow certain desirable functions to be performed across the network. For example, an operating system, or its related utilities, provides for the transfer of a file from one pack to another pack, or from one pack to a tape, connected in the same system. The Host Services extension of that function allows the transfer of files from a pack on one system across the network, to a pack or tape on some different system in the network. Also, just as an operating system allows a program to "read" a record from a file on a disk pack connected to that system, so a "Host Services extension" allows that same program to "read" the record, when in fact the file is located on some other system elsewhere in the network.

Host Services also allows the running of jobs or tasks on remote systems, the logical transfer of terminals between systems, and operator communications between systems. These functions are described more fully later in this Section, and covered in detail in Section 4 of the manual.

In general, existing applications programs require no modification to run in a BNA environment. Simple file equate statements at run-time allow these programs to avail themselves of remote resources.

A feature called Interprocess Communications allows programs at two different hosts in the network to communicate. This feature requires some software modification to take advantage of BNA extensions to the higher level languages. Communications between user programs is discussed more fully later in this section, and in Section 3 of this manual.

Network Services is the interconnect and data transport vehicle that provides communications between processes throughout the network. Network Services provides the capabilities for information to be transferred from any point in the network, to some other point in the network.

In performing these functions, Network Services is responsible for:

1. Determining the best message routing paths.

2. Providing message integrity.

3. Segmentation of messages where necessary.

4. Controlling access to system resources.

5. Keeping of traffic logs and reports.

These functions are, during normal operation, transparent to the user. During the initialization phase of a node, (i.e. when a node is added to the network) Network Services options are selected via a series of attributes which control the characteristics and accessibility of that node to other nodes in the network. Options can be changed during normal network operations as required.

HOST SERVICES

Host Services provides the functions associated with distributed processing in a manner such that distributed processing appears to the user to be as simple as accessing the resources of a single system.

Host Services is a set of host-to-host protocols, one protocol for each Host Services function. A protocol defines a group of messages designed to support the particular function, the intent and format of the messages, and the allowable sequence of the messages. These protocols are supported by Network Services and are not visible to Host Services users.

The functions provided by Host Services include the following:

1) Access by application programs to files located at remote hosts.

2) The movement of files between hosts.

3) The routing of operator messages and system responses between hosts.

4) The logical transfer of terminals between hosts.

5) The transfer of jobs for execution at a remote host.

6) Program and task initiation and control at remote hosts.

New task and file attributes, and additional syntax for user programming languages, control statements, and ODT commands are provided. These allow users to request services at remote hosts in a manner similar to requesting services at a local host.

DESCRIPTION OF HOST SERVICES FUNCTIONS

The following paragraphs describe the six functions provided by Host Services.

Remote File Access

Programs can access files located at remote hosts in the same manner that they access files located at the local host.

As illustrated in Figure 2-2, an application program in host system A can use Host Services to access a file in host system B. The statements required to access the file are the same as if the file were located in host system A, except that the program must indicate that the file is located on another host by setting the file's HOSTNAME attribute to the name of that host. Normal I/O operations are used by the program to create the file, read from it, and update it.

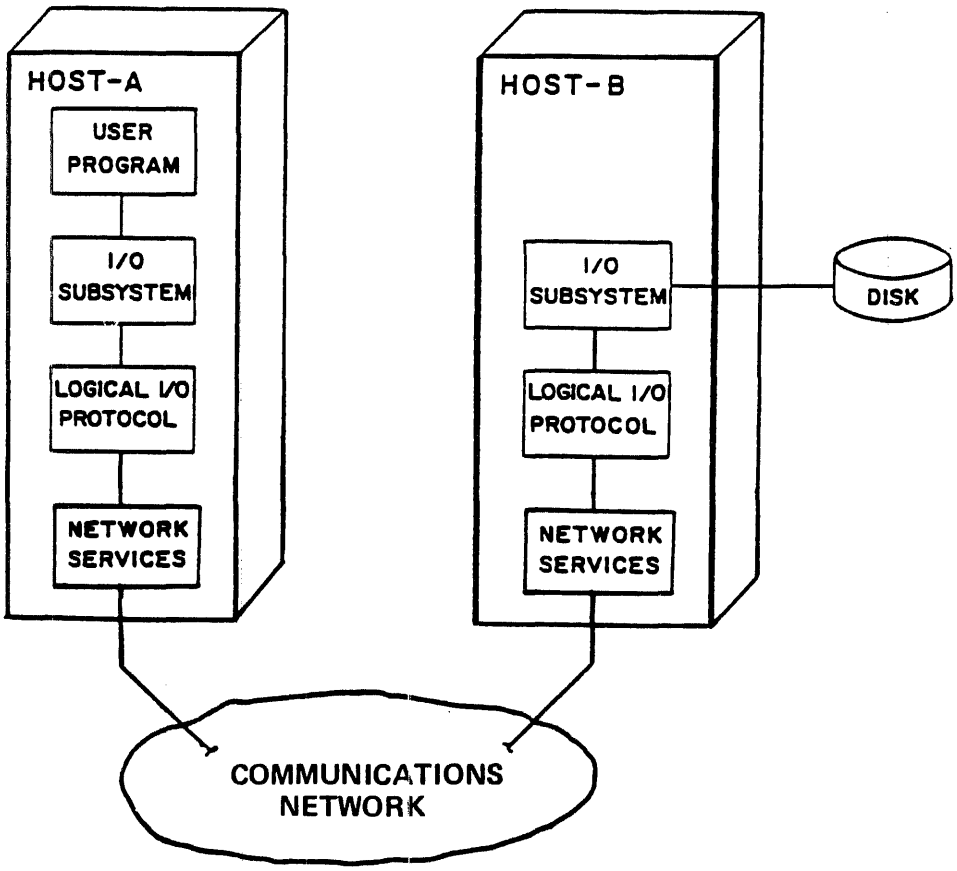


Figure 2-2. Host Services, Remote File Access

File Transfer

An application program or operator can copy files from one host to another in the same way that files are copied from one device to another at a single host.

Files are copied from one host to another by using the HOSTNAME attribute as an extension to the library maintenance functions. An operator or user can copy files from the local host to a remote host by specifying a value for the destination HOSTNAME, from a remote host to the local host by specifying a value for the source HOSTNAME, or from a remote host to another remote host by specifying both.

Figure 2-3 and the following example illustrate the file copy process where a user at the console of host system A can copy a file named "X" from the disk named "P" on host system B to the disk named "Q" on host A in the same manner as the user would copy a file within host A, except that the HOSTNAME for the source file is specified as "B". For example:

```
COPY X FROM P (KIND=DISK, HOSTNAME=B) TO Q (KIND=DISK)
```

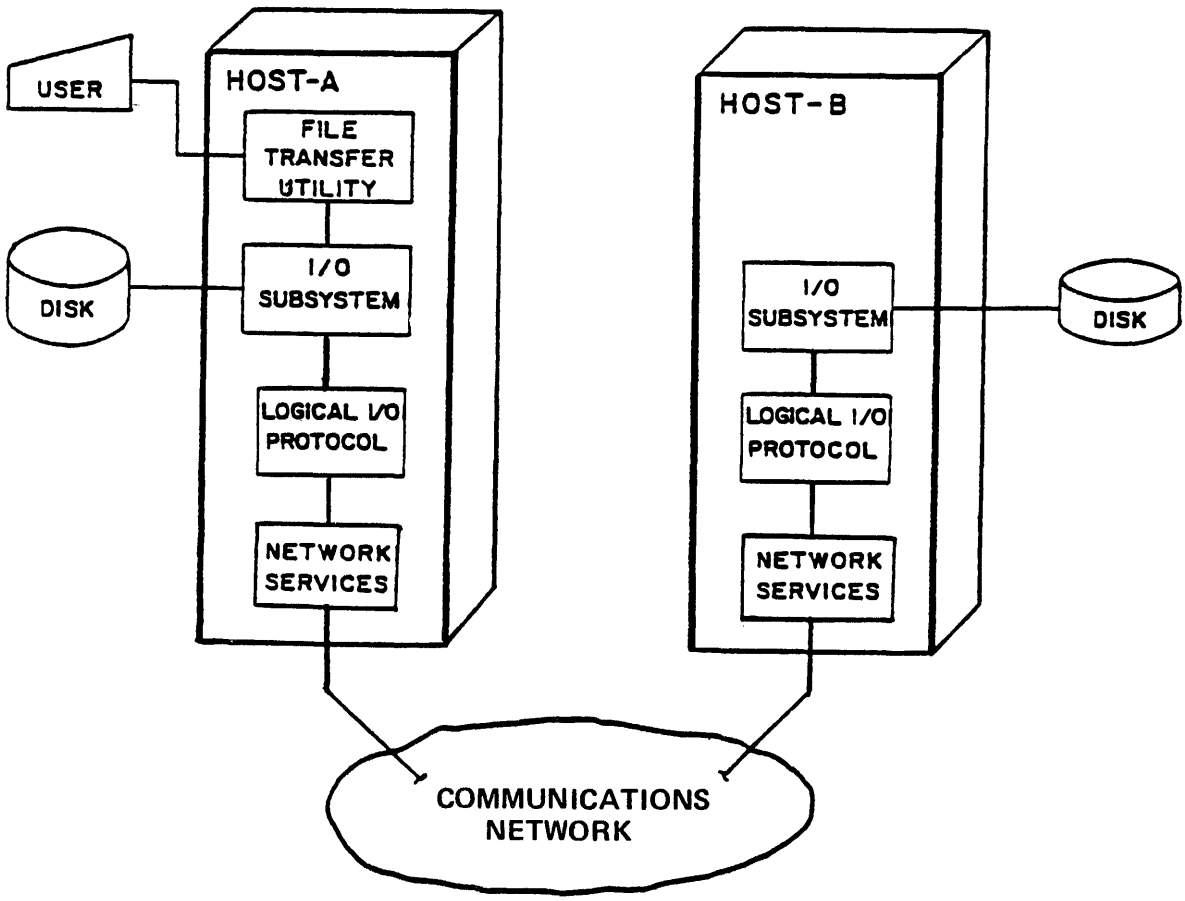


Figure 2-3. Host Services, File Transfer

Operator Inquiry and Control Messages Between Hosts

Host Services allows the operator at the local host to make inquiries and commands to remote hosts concerning either the state of the remote host itself or of tasks running on the remote host. The nature of the inquiries allowable at a remote host is only limited by the access control procedures of the remote host. A remote host's access control procedures can set restrictions on the type of inquiry or command which is accepted from an operator located at a different host.

Operator inputs which are to be delivered to a remote host are prefaced by the phrase "AT <destination hostname>":

AT <destination hostname> <message text>

The message text following the phrase is not checked for syntax by the local host, but is transferred to the remote host specified by <destination hostname>. The remote host examines the text delivered to it for syntactic correctness and access considerations.

If the command or request is correct and allowable, the remote host treats it in the same manner that it would treat a local command or request. It acts upon the text and returns an appropriate response.

igure 2-4 illustrates an Operator Display Terminal (ODT) at host A using the ost Services ODT protocol to perform operator actions at host B.

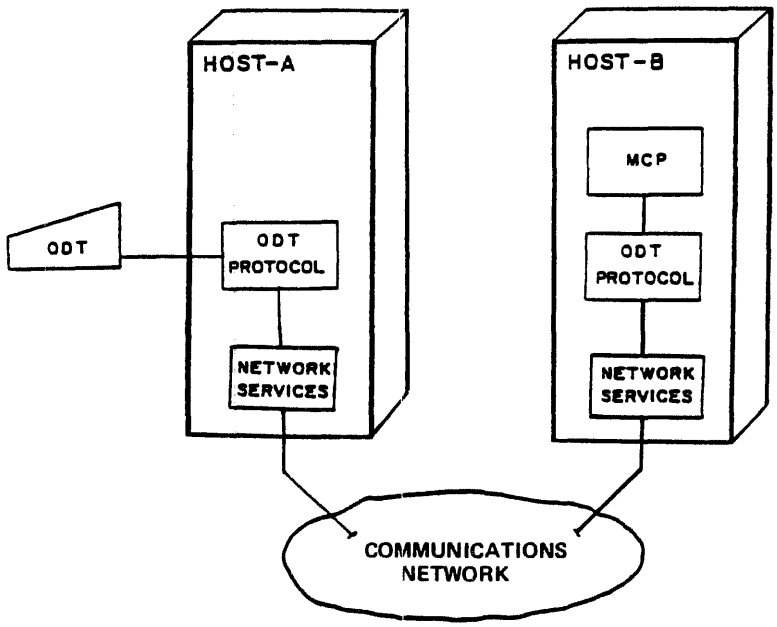


Figure 2-4. Host Services, Operator Inquiry

Logical Transfer of Terminals Between Hosts

Host Services provides the user the ability to utilize existing data communication facilities in conjunction with a BNA network. The station transfer function of BNA provides a means of logically attaching a terminal to a remote host. The user of the terminal appears to have a direct connection to the remote host. The terminal user must be aware that if the remote host is not the same type as the local host, the terminal user must operate in the environment of the remote host. That is, the syntax of the input and output messages are in the format used at the remote host. Physical control of the terminal is maintained by the host to which it is physically connected. Figure 2-5 illustrates a terminal which is physically attached to host A, but logically attached to an application program in host B through the Host Services Station Transfer protocol.

Examples of the syntax to logically attach/detach a terminal to/from program "APPL" at a remote host named "B" are:

```
CONNECT TO B: APPL

DISCONNECT
```

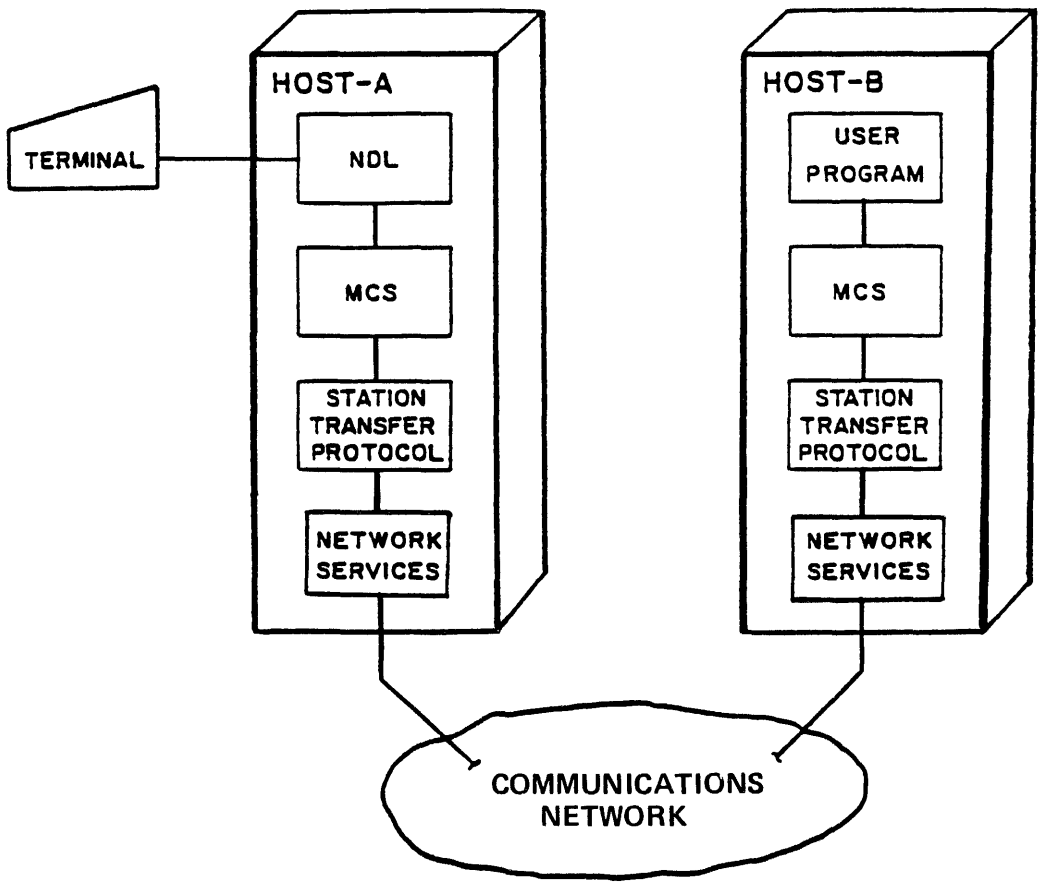


Figure 2-5. Host Services, Station Transfer

Job Transfer

The Host Services Job Transfer protocol is used to transfer a series of job source images from a local host to a remote host for interpretation and execution. The job control statements are not checked for syntax at the sending host, allowing jobs to be transferred between hosts that do not utilize identical control statement syntax. Figure 2-6 illustrates a job transferred from host system A for execution on host system B through the Job Transfer protocol.

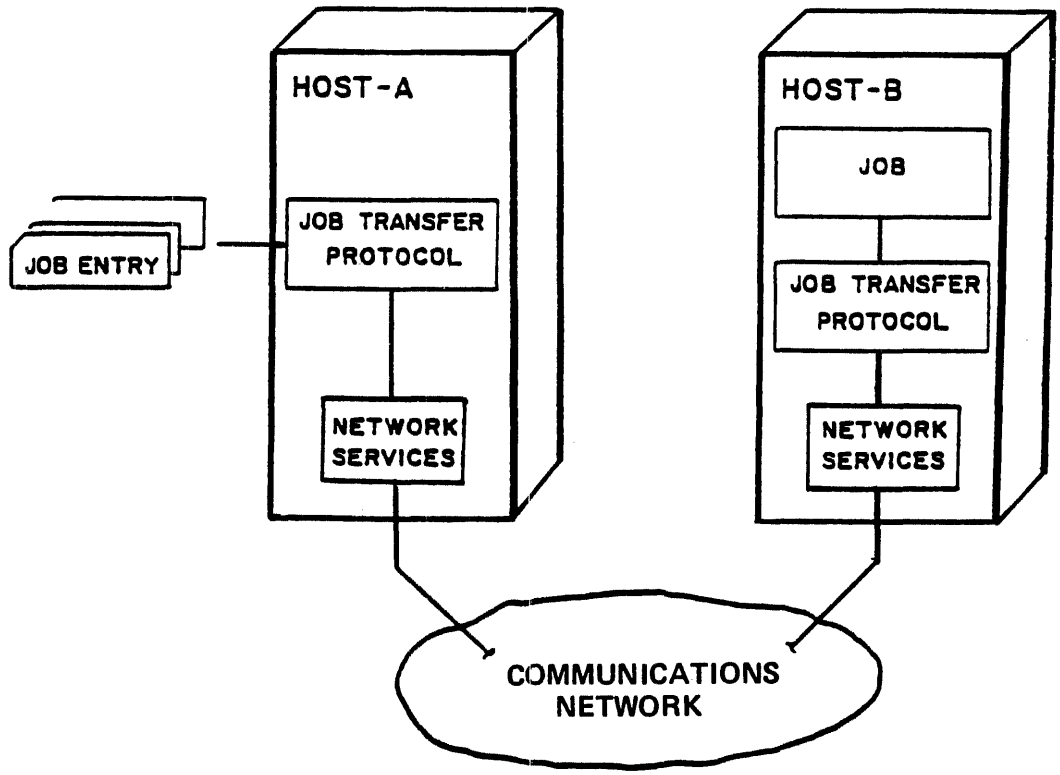


Figure 2-6. Host Services, Job Transfer

Remote Tasking

Some Burroughs products permit a job or task (process) to initiate, monitor, and control "sub-tasks". The Host Services Remote Tasking protocol allows these sub-tasks to be processed at a remote host.

The existing task attributes have been extended to include the HOSTNAME attribute. When the HOSTNAME attribute is set for a task, it indicates the name of the host where the program's code file is to be found and executed. Tasks on remote hosts are initiated in the same manner as tasks on the local host; by Work Flow Language (WFL) jobs, by application programs via a RUN, PROCESS, or CALL of an external code file, and by users of Command and Edit (CANDE) terminals. The program's code file must be resident at the host where it is to be executed, and a usercode must be associated with a job, task, or terminal which does remote tasking. Once a task has been initiated at a remote host, the initiator has the same control capabilities that exist for tasks running on the local host; task attributes can be interrogated and set, and the task can be suspended, resumed and terminated. Figure 2-7 and the following example illustrate an applications job in host A which has initiated a sub-task named "X" in host B through the Host Services Remote Tasking protocol. The code for the subtask is located on the disk named "P" at host B.

yntax example (WFL): RUN X ON P; HOSTNAME=B

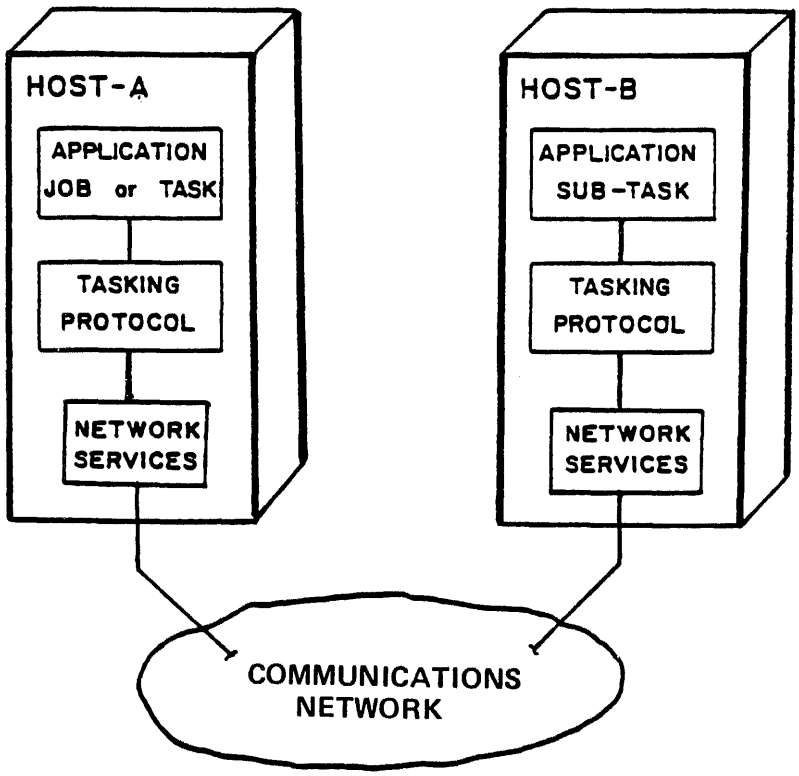


Figure 2-7. Host Services, Remote Tasking

NETWORK SERVICES

Network Services provides the interconnection mechanism for Host Services and for communication between user programs as shown in Figure 2-8.

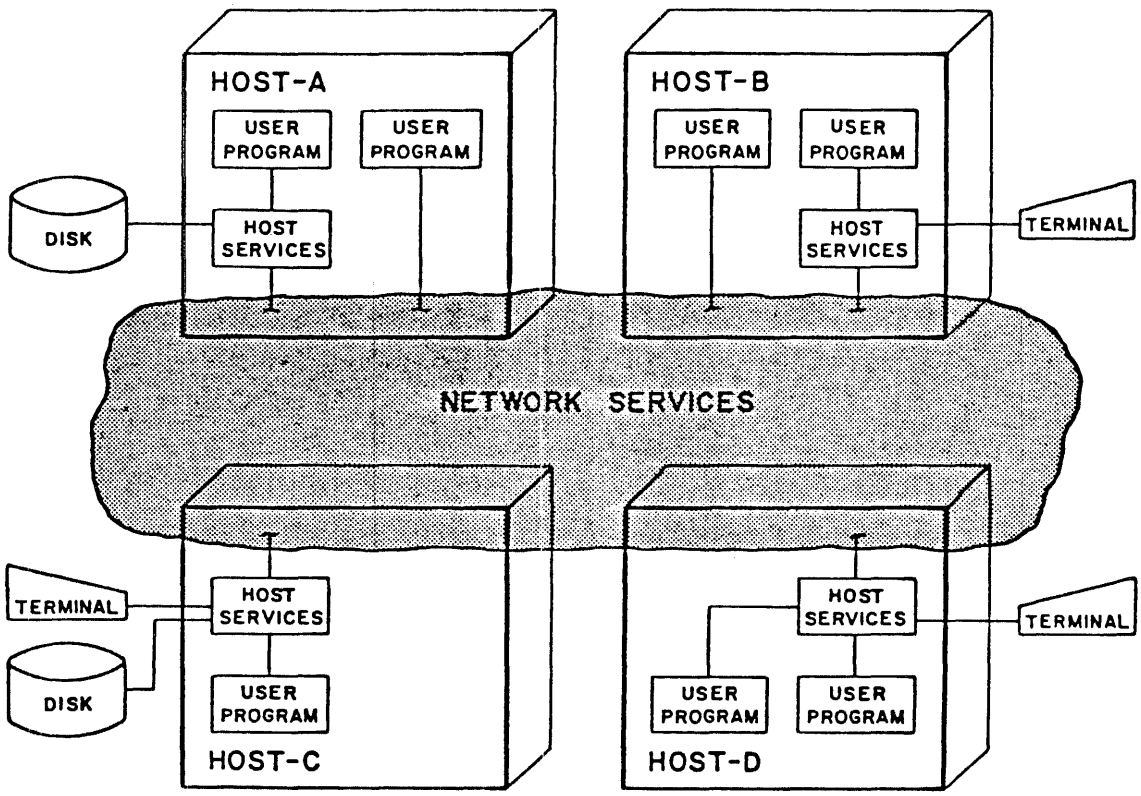


Figure 2-8. Network Services Overview

Within a node, Network Services is functionally divided into three levels and a Network Services Manager. Figure 2-9 shows the structure of a node.

The three levels, from lowest to highest, are: the Station Level, which provides physical link connections between neighbor nodes; the Router Level, which provides a logical connection from each node to every other node in the network; and the Port Level, which ensures reliable transfer of messages from senders to receivers. The Network Services Manager performs management functions to the three functional layers.

At the Station Level, one station exists for each physical link to a neighbor node. At the Port Level, the ports and subports provide the end-points for communications between user programs.

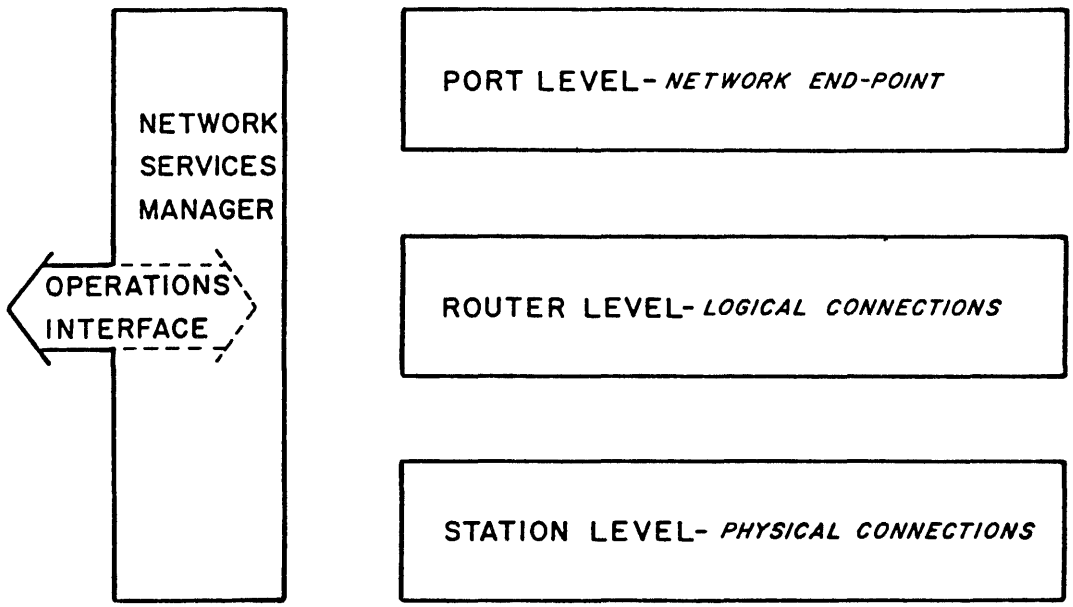


Figure 2-9. Network Services Levels

Each of these levels communicates with the corresponding level at other nodes through the lower levels, thus forming concentric circles of network levels, as illustrated in Figure 2-10.

As this diagram shows, the BNA network can be viewed from the physical interconnection level (forming the STATION level network), from the logical interconnection level (forming the ROUTER level network) or from the user-to-user interface level (forming the PORT level network). These levels are described more fully in this Section, and in detail in Section 5 - Functional Description, Network Services, as the STATION LEVEL, ROUTER LEVEL, and PORT LEVEL, respectively.

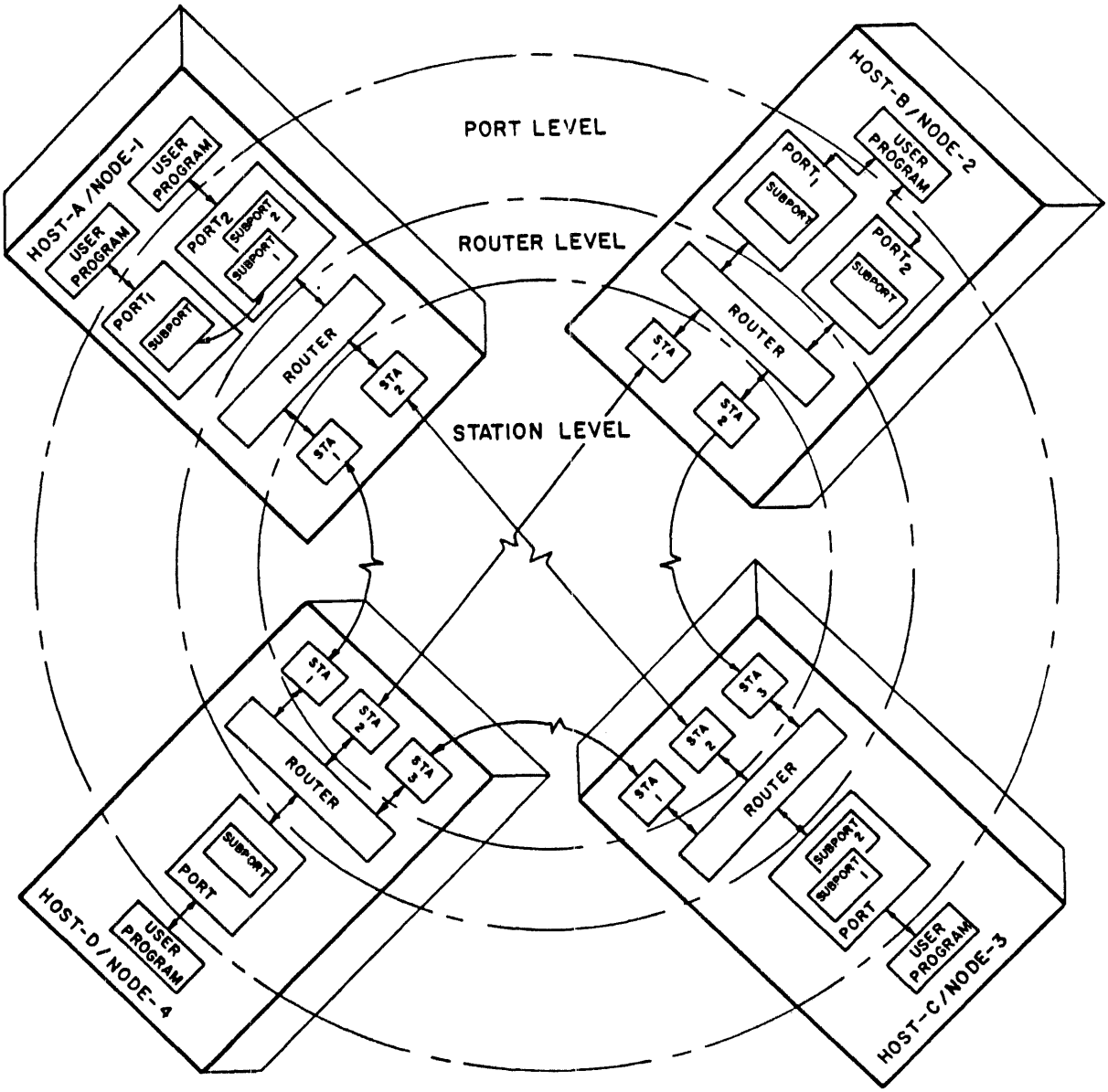


Figure 2-10. BNA Network Levels

PORT LEVEL

The Port level is the highest layer of Network Services. It is responsible for reliably transferring messages from a sender to a receiver. A port can be viewed as an end of the connection path between communicating processes. By performing end-point functions for the connection paths between communicating processes, the port insulates the processes from the idiosyncrasies of message transfer. It provides end-to-end message integrity and a mechanism for establishment of dialogs between users within a BNA network.

In Figure 2-11, the user processes in Host A and Host B are seen connected together via Port 2/Subport 2 of Host A and Port 1 of Host B. The complexities of the lower levels of Network Services are not visible to the user processes or to the ports. The two user processes in Host A are also seen connected together, via Port 1 and Port 2/Subport 1 of Host A. This connection does not use the lower levels of Network Services, but the communications appears the same to the user processes.

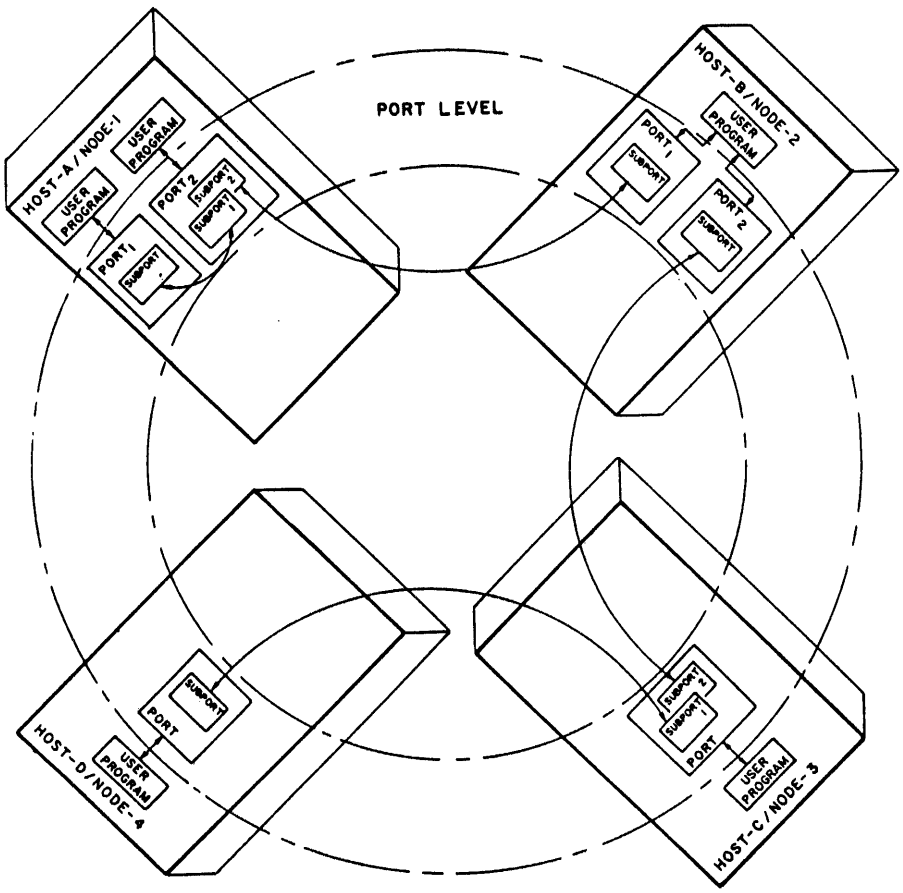


Figure 2-11. Port Level Communications

A port can communicate with one or more remote (or other local) ports. A port has one subport for each port with which it is communicating.

When a process sends a message which is larger than the maximum size allowed by the lower levels of Network Services, the message is segmented by the sending port, and reassembled by the receiving port. It is not reassembled and again segmented at the intermediate nodes. A segment numbering scheme is used to re-order segments for presentation to the receiver. Message segmentation and segment numbering allow the port to ensure message integrity by retransmitting unacknowledged segments and to take full advantage of the effect of multiple parallel links. The receiving port can control the rate of segment transmission.

The Port level controls and coordinates the opening and closing of dialogs between users. It "matches" two users, and negotiates between them to establish a set of compatible communication conditions.

ROUTER LEVEL

The Router Level provides a logical connection from each node to each other node in the network, whether or not there is a physical connection between them.

At the Router level, all Routers are logically connected to all other Routers. As shown in Figure 2-12, the previously discussed connection between Port 2/Subport 2 in Host A and Port 1 in Host B can be made through these logical connections.

At a single node, this level dispatches locally originating traffic to the station level, forwards locally terminating traffic to the port level, and relays transiting traffic to the station level for transmission toward its destination node.

BNA's routing mechanism is referred to as Burroughs Integrated Adaptive-Routing System (BIAS). The BIAS™ mechanism automatically takes into account the capacity of each node for processing transit traffic, and each link's capacity to pass traffic. When there is more than one connection between neighbor nodes, it uses a composite logical link capacity based on the combined capacity of the parallel physical links. These capacities are used to define a "resistance" for each link and for each node. A high capacity link has a low link resistance. A node with greater processing power has a low node resistance. These resistances are used to find the "best" routings across the network. The "best" routing between any two nodes is defined to be the path with the least total resistance (or greatest capacity).

™BIAS is a trademark of Burroughs Corporation.

The BIAS mechanism automatically responds to the following changes in network topology:

1. A node is added to the network.

2. A node is removed from the network.

3. The capacity of an existing node changes.

4. A link is added to the network.

5. A link is removed from the network.

6. The capacity of an existing link changes.

When the network topology changes, the BIAS mechanism automatically redetermines the "best" routings between nodes. Routing control is distributed among the nodes in such a way that each node is responsible for determining the "best" routing to each other node in the network.

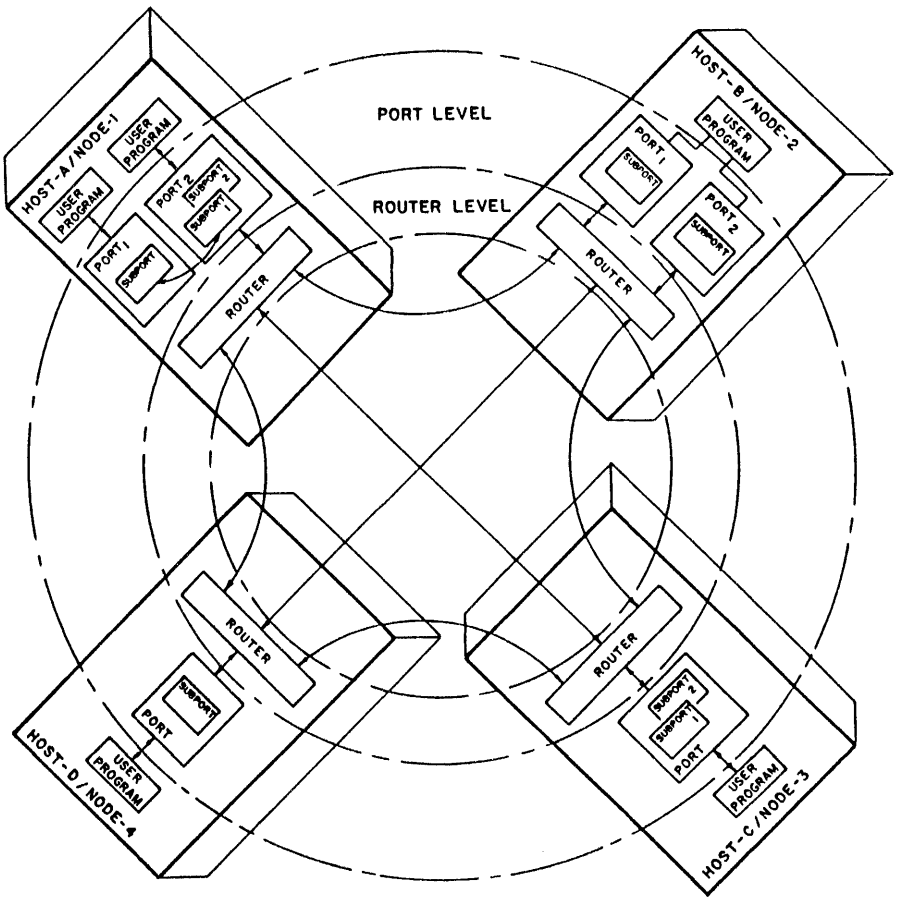


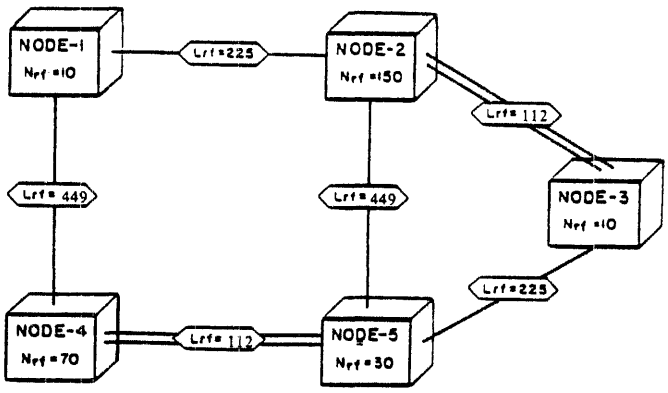
Figure 2-12. Router Level Communications

Figure 2-13 illustrates a single node's perception of the network routing. The routing tables within Node 1 list all of its choices of routes to each of the other nodes (Destination Node Addresses, or DNA) in the network. To each, it has two choices, one through each of the nodes to which it is physically connected, neighbor nodes (NNA) 2 and 4. For each choice, its tables contain the total resistance to each destination node. The total resistance is the sum of the total link resistances (Lrf) and node resistances (Nrf) along the path to the destination node. The active routings are the routing choices with the lowest total resistance to each DNA.

The active path to Node 5 is via Neighbor Node 4. It has a total resistance of 631, far better than the 824 resistance that would be encountered if the path through Neighbor Node 2 was selected. The resistance of 631 is the total of 449 (the Node 1 - Node 4 link resistance) + 70 (Node 4's internal resistance) + 112 (the Node 4 - Node 5 link resistance). The node resistances at the end-points are not counted.

The BIAS mechanism does not use pre-defined, pre-loaded, or pre-coordinated routing tables. The routing tables are built by each node as the network evolves. To help the router in each node build its tables, a minimal amount of information must be supplied when the node is initialized. The items which must be supplied are the node's own address, its resistance factor, and the speed and efficiency factors for the links connecting it to its neighbors.

The Router level also provides two support functions for network operational analysis, Trace and Monitor. Trace is used to confirm the routing path between network nodes. Monitor is used to record various Router occurrences for analysis of network operation.



NODE 1'S PERCEPTION OF NETWORK

DNA	NNA	RESISTANCE	
2	2	225	(ACTIVE)
	4	1008	
3	2	487	(ACTIVE)
	4	886	
4	2	864	
	4	449	(ACTIVE)
5	2	722	
	4	631	(ACTIVE)

Figure 2-13. Network Routing Example

STATION LEVEL

The primary purpose of the Station Level is to provide error-free data transmission between the neighbor nodes. To this end, each station is paired with another station in another node, either temporarily or permanently. The Station Level consists of a Station Level Manager and one or more stations. A pair of stations and the communication links between them provide the physical interconnection of two nodes in a BNA network. A pair of interconnected nodes are referred to as "neighbor nodes".

Figure 2-10 illustrates the Station level network along with the Port level and Router level networks. Each station is connected to only one other station. At this level, the physical interconnection level, the diagram shows that all nodes are not necessarily interconnected. The Router level's logical connection between Nodes 1 and 2 is not supported by a direct physical connection, but by a multiple-hop connection through Nodes 3 and/or 4.

Neighbor nodes can be connected by dedicated (leased) or switched point-to-point telephone links or by Public Data Networks (PDNs) which support the International Telephone and Telegraph Consultative Committee (CCITT) X.25 interface. The following types and variations of Stations are supported in BNA:

- Burroughs Data Link Control (BDLC)
  - Dedicated (Leased)
  - Switched
    - Manual Answer (Dial-in)
    - Auto Answer (Dial-in)
    - Manual Dial-out
    - Auto Dial-out (via ACU)
- X.25
  - Permanent Virtual Circuit
  - Virtual Call (Incoming)
  - Virtual Call (Outgoing)
- Global Memory ™

™Global Memory is a trademark of Burroughs Corporation.

Two neighbor nodes can be configured with multiple connections between them. This capability provides better reliability through the use of active backup links, increased bandwidth between nodes, and generally decreased response time as seen by network users. Each of the links between a pair of neighbor nodes operates independently, which allows a mixture of link types (dedicated, switched, packet switched) to be used in parallel between nodes. Segments that arrive out of order are allowed to pass that way to the Router. Reordering is accomplished in the Port Level when these segments are reconstructed into messages. This facility allows the user to augment dedicated links with switched connections to create larger capacity during peak traffic periods. It also provides uninterrupted service when the user switches from one communications facility to another whether for economic reasons, capacity requirements, or facility maintenance.

BDLC Station

Burroughs Data Link Control (BDLC) is used as the link-level communications protocol between BNA nodes which are connected via point-to-point telephone links. BDLC is a bit-oriented protocol developed by Burroughs. It is similar to High Level Data Link Control (HDLC) and ADCCP (Advanced Data Communications Control Procedure).

When full-duplex dedicated (leased) links are used, BDLC operates in a two-way simultaneous mode. When half-duplex switched links are used, a two-way alternate mode of operation is used.

The BDLC station supports the CCITT V.24 and Electronic Industries Association (EIA) RS232C interface to datasets and the CCITT V.25 and EIA RS366 Auto-Call Unit (ACU) interfaces. The node interconnections in Figure 2-14 and in Figure 2-10 are examples of BDLC point-to-point circuits.

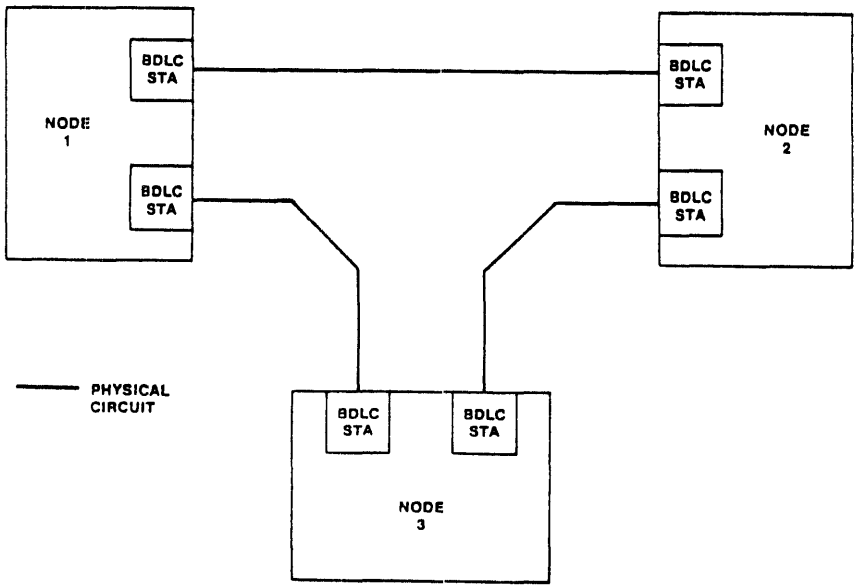


Figure 2-14. BDLC Station Interconnections

X.25 Station

Depending on network usage and line costs, using a Public Data Network (PDN) to connect BNA nodes can be more economical than using point-to-point telephone circuits. When neighboring BNA nodes are connected by a PDN, they interface to the network according to CCITT Provisional Recommendation X.25 (Interface between Data Terminal Equipment (DTE) and Data Circuit - Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks).

The X.25 interface defines a means of multiplexing multiple, independent traffic-streams over one physical circuit between node and PDN. In principle, the PDN is able to fully interconnect a network of up to 4095 nodes with each node supporting only one physical circuit. These nodes are interconnected not by physical circuits but by "virtual circuits". Figure 2-15 illustrates this kind of interconnection.

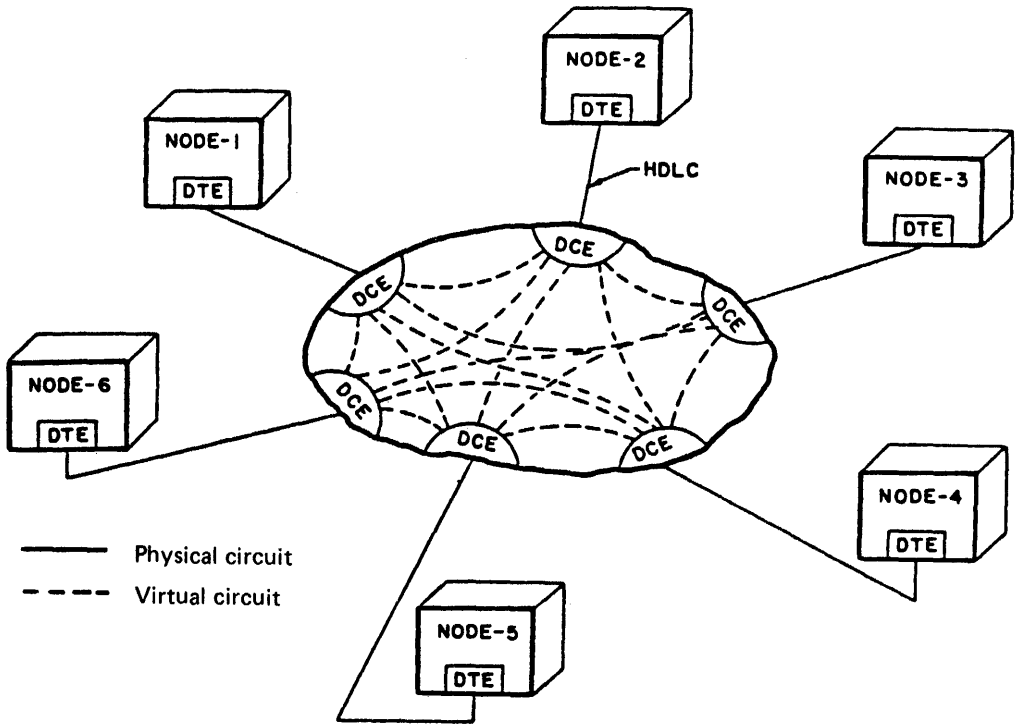


Figure 2-15. X.25 Station Interconnections

The complete CCITT Provisional Recommendation X.25 is defined by the International Telecommunication Union in Geneva, Switzerland. Information on a particular PDN's implementation of X.25, its service offerings, and its tariff structure should be obtained from the vendor of that X.25 service.

The BNA X.25 station provides the DTE functions to support these facilities:

- At the Packet Level: virtual calls (VC) and permanent virtual circuits (PVC). PVCs are similar to common dedicated or leased telephone circuits insofar as a connection is always present between the two end points (i.e., the two nodes). Virtual calls (VC) are similar to common dialed telephone circuits insofar as a connection between two end points is not always present and must be established and ultimately cleared according to a defined procedure.

- At the Link Level: LAP B (balanced) using dedicated full-duplex circuits between the DTE (BNA node) and the DCE (Public Data Network).

- And at the Physical/Electrical Level: CCITT X.21bis, CCITT V.24, and EIA RS232C.

NETWORK SERVICES MANAGER

The Network Services Manager performs management functions which are global to the three functional levels of Network Services as illustrated in Figure 2-9. These include:

1. Network Services initialization.

2. Monitoring and logging.

3. Management of interlevel control communications.

4. Providing an operations interface for Network Services.

5. Disconnecting the node from the network.

NETWORK SERVICES FRAMES

Information passes between the three major levels - PORT, ROUTER, and STATION - in units called Network Services FRAMES, or simply FRAMES. These are named according to which functional level in the host generates or interprets them.

Those frames interpreted by the Station are called LINK FRAMES.

Those interpreted by the Router are called ROUTER FRAMES.

Those interpreted by the Port Level are called PORT FRAMES.

These frames are organized in a hierarchical manner. For example, the Port Frame is passed to the Router, which then treats this frame as a single field (called the ROUTER INFORMATION UNIT), and adds a header to it. The result is a ROUTER FRAME, which in turn is passed to the Station Level. This structure of a frame within the body of another frame is illustrated in Figure 2-16.

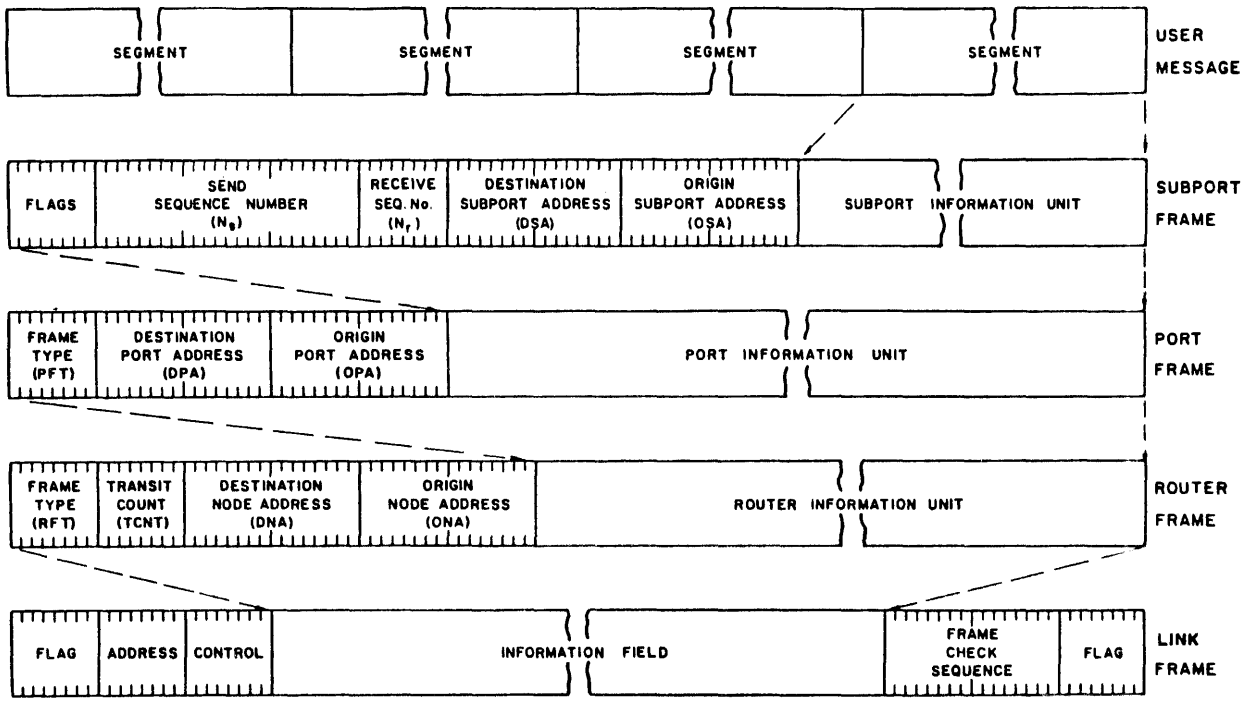


Figure 2-16. Frame Format Relationships

Use of Frames to Transmit User Data

A user process (or Host Services) sends a message by passing it to the Port level of Network Services. Port and subport headers are attached to form a port frame. This is passed down through the ROUTER and STATION level as described above, and as illustrated in Figure 2-17. The message leaves the sending node as a link frame, which is sent out over the communication link. At the destination node, it is received by the STATION level and passed up through the ROUTER and PORT levels. Each level strips off the appropriate headers, restoring the message to its original form. It is then placed into the port's output queue and made available to the destination user process (or Host Services).

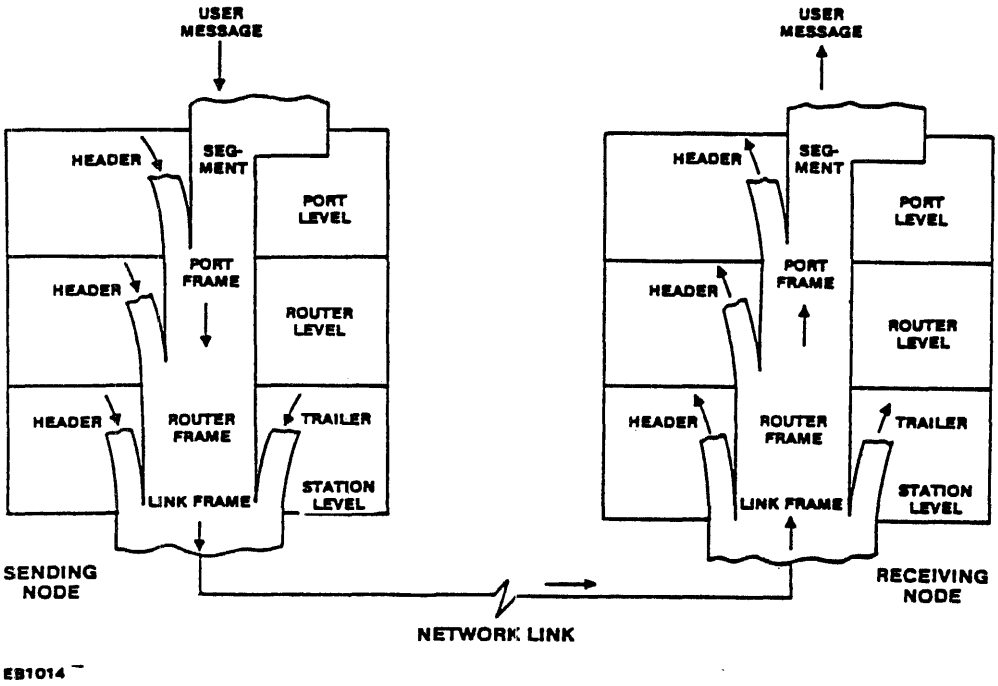


Figure 2-17. User Message Transmission

Use of Frames to Transmit Control Data

If the information contained in a frame is not originated by a Port user, then it is originated by one of the three levels - the PORT, ROUTER, or STATION. Refer to Figure 2-18.

1. If the Port Level originates the information, it forms what is called a PORT LEVEL MANAGER (PLM) CONTROL FRAME. This is interpreted by the corresponding destination Port Level. This control information is used, for example, when the local Port Level is communicating with a remote Port Level, or when the local subport informs the corresponding remote subport that it wishes to terminate its dialog. The PLM CONTROL FRAME is treated the same as any other Port Frame in the Router. To the Router, it is "user data" regardless of where or why it was originated in the Port level above it. Therefore, the PLM CONTROL FRAME is handled the same as the Port user data at the Router Level and below, both at the originating and the destination nodes (and at any transit nodes).

2. If the Router originates the information, it is called a ROUTER CONTROL FRAME. These frames are generated, for example, whenever the Router sends routing change information to a neighboring node. ROUTER CONTROL FRAMES are interpreted by the corresponding Router at the remote node. Except for priority, the Station treats this frame the same as any other frame received from the Router. It is "user data" to the Station, both at the originating and destination nodes.

3. Finally, if the Station Level generates a control frame, it is interpreted by the corresponding Station Level at the other end of the link, and goes no higher in the node. STATION LEVEL MANAGER (SLM) CONTROL FRAMES are generated, for example, for Station Level validation greetings and as test frames for maintenance purposes.

OPERATIONS INTERFACE TO NETWORK SERVICES

The Operations Interface provides a mechanism for the exchange of messages between Network Services and an external agent, which can be either a human operator at an operator display terminal (ODT) or a user-defined program. Additionally, operating parameters can be loaded into Network Services from a disk file.

Uses of the Operations Interface include initializing Network Services, changing operating parameters, adding or deleting links, monitoring operations, performing confidence tests, and initiating node disconnection from the network. BNA Reference Manual, Volume 2 (Network Control) discusses the Operations Interface commands and responses.

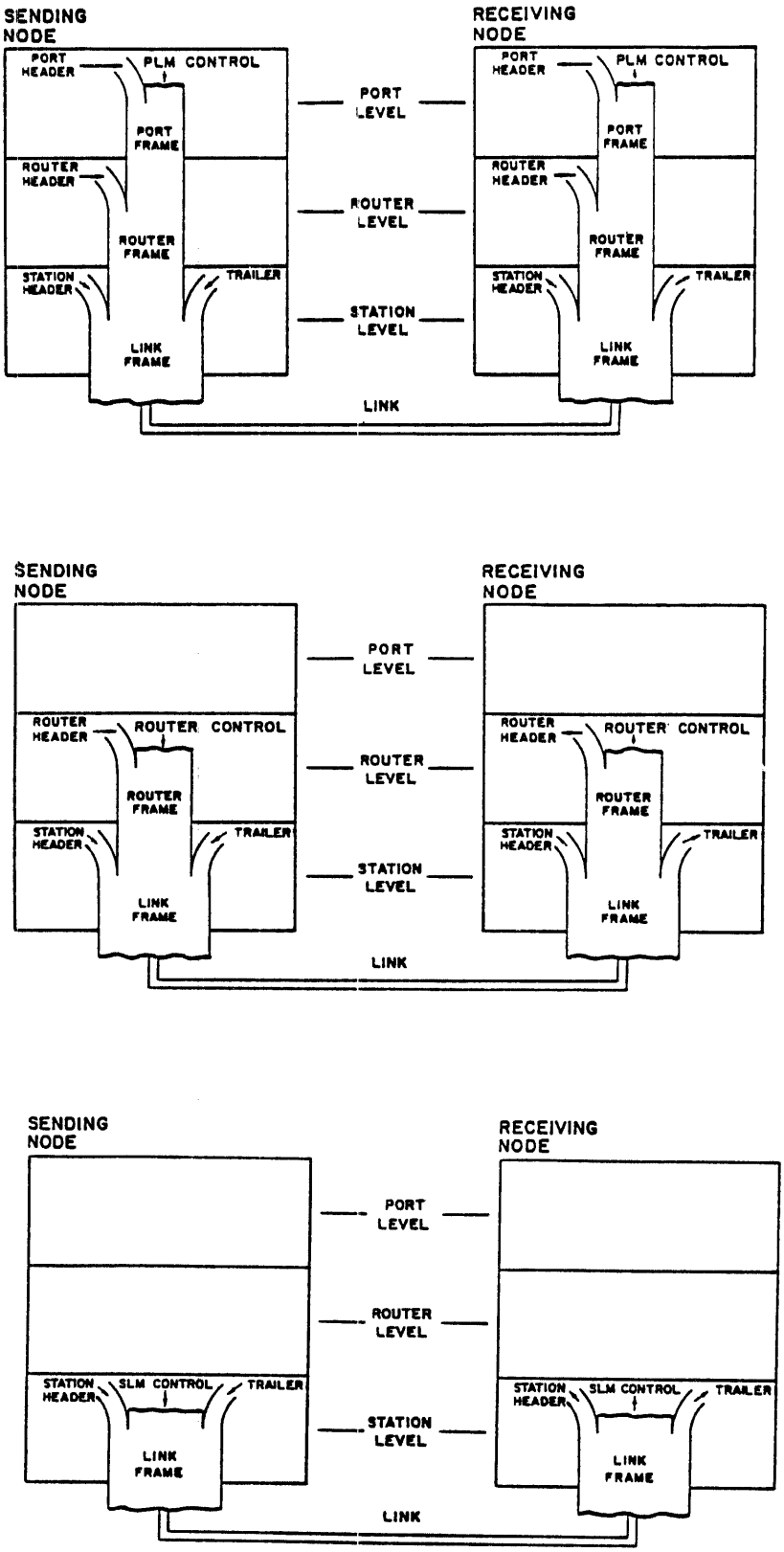


Figure 2-18. Control Message Transmission

COMMUNICATION BETWEEN USER PROGRAMS

BNA supports a general inter-process communications mechanism, using the Port layer of Network Services. Refer to Figure 2-11. User programs can communicate on a one-to-one basis, or one-to-many basis or on a many-to-many basis. The user programs can be at the same host, or at any host in the network.

Communication between user programs across a BNA network (or locally) is accomplished through the input/output file mechanism using a special kind of file called a PORT file.

A user program can communicate with a remote user program by performing WRITE and READ operations to a port file. The remote user program must perform complementary READ and WRITE operations. A port file has one or more associated subfiles, each of which can be connected to a different program. These programs can be located at any host within a BNA network.

Figure 2-19 illustrates the relationship between hosts, user programs, port files, and subfiles. Note that each subfile is connected to only one other subfile. Also note, in Host H, the example of a local connection between two user programs in the same host.

BNA defines additions and extensions to several file attributes which return information about the port file and its subfiles. There are four basic operations which can be performed on port files: OPEN, CLOSE, READ, and WRITE. These are discussed in the following paragraphs. More detailed information on communication between user programs is covered in Section 3 of this manual.

OPEN Operations

An Open operation on a subfile causes the system to find a remote endpoint that matches the subfile and to set up a dialog with that matching subfile.

A pair of subfiles provides a two-way, point-to-point logical communication path between two programs. In order to establish this path, both programs must describe the desired connection. The system compares connection descriptions, matches complementary descriptions, and marks the subfiles opened. This process is called the matching algorithm.

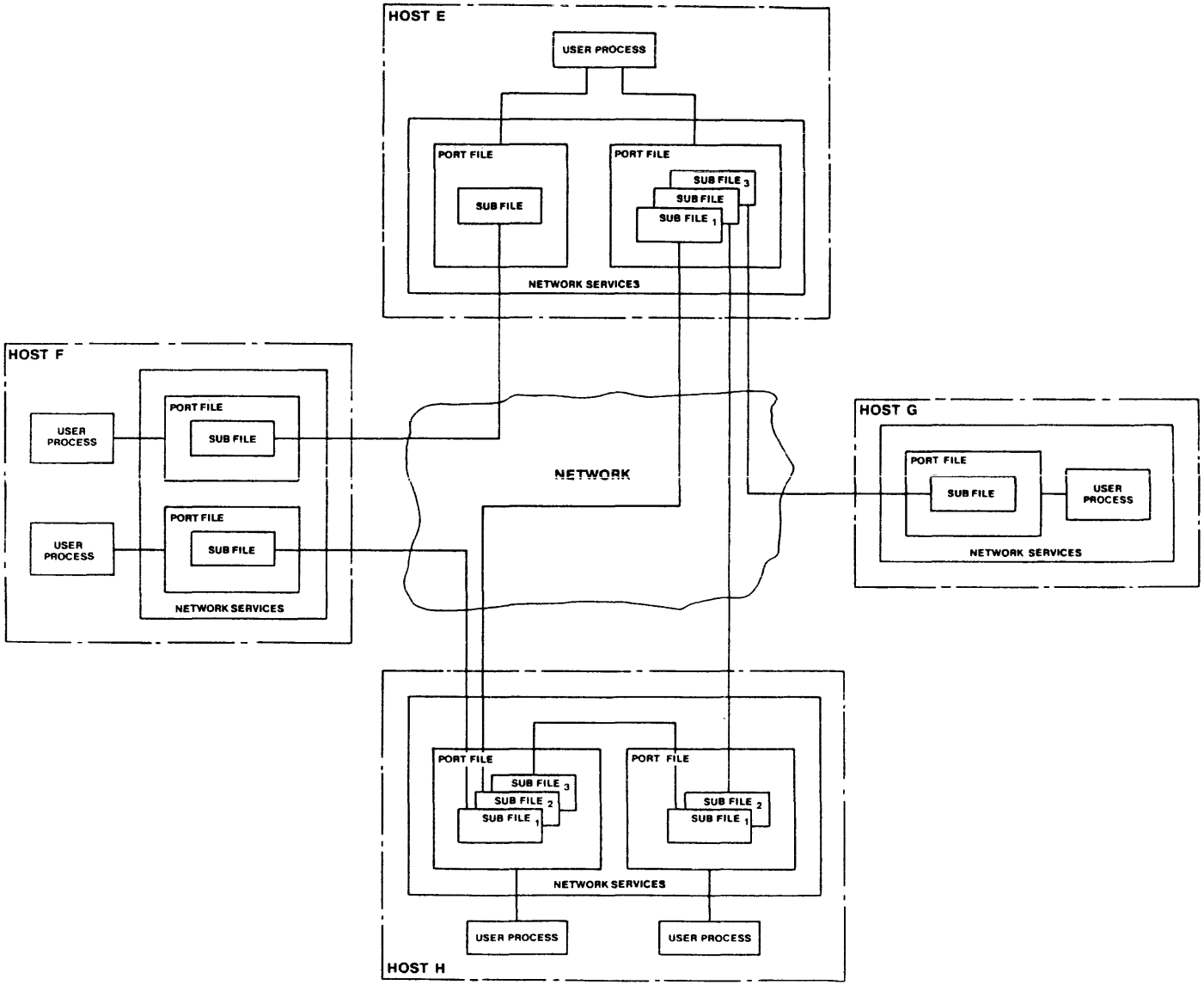


Figure 2-19. File/Subfile Interconnection

The OPEN statement includes the file/subfile that is to be opened and an open option. The processing of the OPEN statement causes the matching algorithm to be invoked and causes a result to be returned indicating the success or failure of the open attempt. The subfile to be opened is specified by a subfile index.

The following three open options apply to port files:

1. WAIT
   The subfile is offered for matching, and the program is suspended until a matching subfile is found. WAIT is the default value.

2. OFFER
   This option causes the subfile to be offered for matching, and the program is resumed without waiting for the subfile to be matched. The program can determine when the subfile is open by monitoring state changes.

3. AVAILABLE
   This option causes the subfile to be matched only to a complementary subfile that has been already offered. If a match is found, the subfile is opened. If not, the program is resumed and the subfile is not left waiting for a match.

Connections between programs wishing to communicate are made based on attributes of the port declared by the program. Attributes used to determine the connection are PORTNAME, MYNAME, YOURNAME, YOURHOSTNAME, and YOURUSERCODE, as well as the local hostname and usercode. A program can have many ports open and can communicate simultaneously with programs on many hosts.

Connections between ports are subject to security checking similar to file security. Ports can be PRIVATE, PUBLIC, or GUARDED.

CLOSE Operations

The Close operation terminates the dialog with the remote subfile and returns the system resources used by the local subfile. If the close option DONTWAIT is specified, control returns immediately to the program and the process of actually closing the port takes place in parallel with the execution of the program. The program can detect when the close is complete by monitoring state changes.

I/O Operations

A program can perform a Read or Write to a file. Each subfile has a unique subfile index and a program can, by specifying a subfile index, perform a read or write operation to a particular subfile. The subfile index can be ZERO (meaning any or all) to read the next message from any subfile or to broadcast to all open subfiles within a file.

If the DONTWAIT option is specified in the I/O statement, the program continues and the I/O operation is terminated when the I/O operation cannot be completed immediately.

BNA Extensions to Programming Languages

The following paragraphs briefly describe extensions to the various programming languages which are provided to facilitate the use of Port files. Refer to the appropriate systems language manual for specific details of the constructs, syntax, restrictions, etc.

ALGOL

ALGOL recognizes the value PORT for the KIND attribute and recognizes all the additional BNA file attributes and values. To provide access to the port open options, an OPEN statement has been added to the syntax. The CLOSE statement has been modified to allow selection of a subfile and to allow DONTWAIT as a close option. Subfile specification and the DONTWAIT option have been added to the READ/WRITE statement syntax.

FORTRAN 77

FORTRAN77 recognizes the value PORT for the KIND attribute and most of the additional BNA file attributes and values. To provide access to the port open options, the OPEN statement has been modified. The CLOSE statement has been modified to allow selection of a subfile. Subfile specification has been added to the READ/WRITE statement syntax. The INQUIRE and CHANGE statements have been extended to allow reading and writing of subfile attributes.

PL/I

PL/I recognizes the value PORT for the KIND attribute and recognizes all the additional BNA file attributes and values. To provide access to the port open options, the OPEN statement has been modified. The OPEN and CLOSE statements have been modified to allow selection of a subfile. Subfile specification and the DONTWAIT option have been added to the READ/WRITE statement syntax. The WAIT statement has been added to handle waiting on event-valued port file attributes.

COBOL 74

COBOL74 recognizes the value PORT for the KIND attribute and recognizes all the additional BNA file attributes and values. Files are declared by assigning to PORT, and an optional Actual Key clause can be declared which can be used to reference subfiles in the OPEN, CLOSE, and READ/WRITE statements. To provide access to the port open options, the OPEN statement has been extended. The DONTWAIT option has been added to the READ/WRITE statement syntax. The WAIT statement recognizes the event-valued port file attributes relevant to port files. Subfiles are designated as parenthesized expressions following the file name in ATTRIBUTE statements.

NETWORK SERVICES INITIALIZATION

Since control is a distributed function in a BNA network, responsibility for network initialization, both at the beginning and as nodes are added to the network, lies with each participating node. When a node enters the network, it establishes communication with other nodes in the network through a series of greeting messages at the various levels of Network Services.

The Station level is the first level to be initialized. Neighbor nodes establish the physical interconnections between themselves, validating each other's identity through the use of node addresses and passwords.

After a node which is new to the network or previously isolated from the network establishes physical connections to its neighbors at the Station Level, logical connections to all nodes are established at the Router Level. Router messages are exchanged among neighbor nodes, informing one another of paths to newly added nodes, or improved paths to existing nodes as a result of new node participation in the network.

When a node's Router discovers a new node, its Port Level establishes communication with the new node's Port Level by exchanging Port Level greetings. These messages contain information about the host system's identity and willingness to communicate.

After Network Services initialization is completed, it can be used to support distributed processing among hosts.

Additional information on Initialization is covered in Section 6 of this manual.

ACCESS CONTROL

Access control in a BNA network is the means by which each node and host can control access to its resources from nodes, hosts, and users elsewhere in the network. Since hosts in a BNA network are peers, access control in a BNA network is a distributed function. Each node and host is responsible for controlling access to its own resources. Extensive options permit the Network Operations Manager at each node/host to choose the types and amounts of protection for that node/host. There are two major types of access control:

Control of access by other nodes and hosts.

Control of access by individual users at those hosts.

Access by other nodes and hosts is controlled by validation and authentication. Validation controls which nodes and hosts are allowed access. Authentication verifies that a node or host really is the node or host that it claims to be. Access by individual users is controlled on the basis of a Hostname/Usercode pair. The usercode is authenticated at the remote host when the user logs on to the remote host with the usercode/password by which the user is known at that host.

The various validation functions permit the Operations function at a node to pre-define which hosts, nodes, and neighbors are acceptable hosts, nodes, and neighbors. Validation is optional and can be performed at all 3 levels of Network Services; Port, Router, and Station. Validation is applied when communications are initiated with a node/host.

The node and host authentication functions verify that a node really is the node it claims to be by comparing the received password with a pre-stored password for that node. Password checking can be performed at the Station Level between neighbor nodes, and at the Port Level during the establishment of dialogs between hosts. Like validation, authentication is applied when communication is begun with the remote node/host.

Individual users at a host are known elsewhere in the network by their hostname/usercode pairs. A hostname/usercode pair is unique in the network. Even if users on two different hosts have the same usercode, they have different hostname/usercode pairs because the hostnames are different. Users on the same host in the network have different hostname/usercode pairs because their usercodes are different. The hostname/usercode pair is used to check a remote user's access privileges using the same mechanism that is invoked for local usercodes.

An individual user at a remote host seeking access to a subport uses his hostname/usercode pair to establish a connection to that subport. It is also used in the initiation phase of Host Services protocols.

Additional information on Access Control is covered in Section 7 of this manual.

SECTION 3

COMMUNICATION BETWEEN USER PROCESSES

GENERAL DESCRIPTION

Communication between user processes across a BNA network (or locally) is accomplished through the input/output file mechanism using a special kind of file called a PORT file. A user process communicates with a foreign process by performing READ and WRITE operations to a file of KIND = PORT. The foreign user process must perform complementary WRITE and READ operations. File attributes, parameters to the Open and Close functions, and other language constructs provide access to the inter-process communication facility. This section describes the general port file facility, the attributes of a port file, the OPEN/CLOSE, READ and WRITE statements associated with a port file, and provides sample programs which show the use of port files by a user program. Refer to the appropriate systems programming language manuals for details of the syntax to be used with port files. Refer to Section 5 of this manual for a description of the workings of the port and subport.

FILE ATTRIBUTES

A port file has one or more associated subfiles (called "subports"), each of which may be connected to a different process. These processes may be located at any host within a BNA network. File attributes contain information about the port file or its subfiles. The following table describes the file attributes applicable to port files. Attributes marked "File" pertain to the port file as a whole, while attributes marked "Subfile" apply to each individual subfile of the file.

File/Subfile Attribute	Port/Subport Attributes	
BLOCKSTRUCTURE		File
CENSUS	MESSAGE QUEUE SIZE	File,Subfile
CHANGEDSUBFILE	CHANGED SUBPORT	File
CHANGEEVENT	CHANGED EVENT STATE EVENT	File Subfile
COMPRESSION	SENDING COMPRESSED DATA	Subfile
FILESTATE	SUBPORT STATE	Subfile
FRAME SIZE		File
HOSTNAME	YOUR HOST NAME	Subfile
INPUTEVENT	INPUT EVENT	File,Subfile
INTNAME	INTNAME	File
LASTSUBFILE	LAST SUBPORT USED	File
MAXCENSUS	MESSAGE QUEUE LIMIT	Subfile
MAXRECSIZE	MAX MESSAGE TEXT SIZE ACTUAL MAX MESSAGE TEXT SIZE	File Subfile
MAXSUBFILES	MAX SUBPORTS	File
MYHOSTNAME	MY HOST NAME	File
MYNAME	MY NAME	File
OUTPUTEVENT	OUTPUT EVENT	Subfile
SECURITYGUARD	SECURITY GUARD	File
SECURITYTYPE	SECURITY TYPE	File
STATE		File
SUBFILEERROR	SUBPORT ERROR	Subfile
TITLE	PORT NAME	File
YOURNAME	YOUR NAME	Subfile
YOURUSERCODE	YOUR USERCODE	Subfile

Attributes that are only file attributes can be accessed or assigned by not specifying a subfile index. If a subfile index is specified for a file attribute access or assignment, an attribute error is generated.

Those attributes which are only subfile attributes can be assigned for a particular subfile by providing a subfile index. If a subfile index of zero is specified, the attribute assignment applies to all subfiles in the file. If MAXSUBFILES is equal to one, the subfile index may be omitted; the attribute assignment will apply to the only subfile.

When accessing a subfile attribute of a particular subfile, the subfile index must be specified if MAXSUBFILES is greater than one. If MAXSUBFILES is equal to one, the subfile index may be omitted; the attribute access will apply to the only subfile. If a subfile index of zero is specified for an attribute access an attribute error is generated.

For attributes that are both file and subfile attributes, if a subfile index is not specified, the attribute access or assignment applies to the file; otherwise the attribute access or assignment applies to the subfile.

The null value for all string valued attributes is ".".

The following attribute descriptions apply to port files:

BLOCKSTRUCTURE
: The BLOCKSTRUCTURE values of FIXED and EXTERNAL apply to port files. The default is FIXED. BLOCKSTRUCTURE is meaningful only for READ operations. If BLOCKSTRUCTURE is equal to FIXED, the user's buffer is blank filled. If BLOCKSTRUCTURE is equal to EXTERNAL, only the data received is put into the user's buffer. The actual length of the data placed into the user's buffer can be determined by interrogating the CURRENTRECORD attribute.

CENSUS
: The CENSUS attribute can be accessed but not assigned. If accessed as a file attribute, CENSUS returns the total number of incoming messages queued for subfiles. If accessed as a subfile attribute, CENSUS returns the number of messages queued for the specified subfile.

CHANGEDSUBFILE
: CHANGEDSUBFILE is an access-only attribute that returns the subfile index of an arbitrary subfile whose CHANGEEVENT is "happened".

CHANGEEVENT
: The subfile CHANGEEVENT is caused whenever the value of FILESTATE changes. It is reset as a side effect of interrogating the FILESTATE attribute. The CHANGEEVENT for the file has the value "happened" as long as any of the subfile CHANGEEVENTs have the value "happened". The CHANGEEVENT for the file is reset by the system after all of the subfile CHANGEEVENTs have been reset.

COMPRESSION

It is possible to compress the data sent between subfiles. Support for the compression feature is negotiated at subfile open time. Setting the value of the COMPRESSION attribute to TRUE while the subfile is open has an effect only if both hosts involved in the subfile dialog support the compression feature (if compression is not supported, the value of COMPRESSION will be FALSE even after setting it to TRUE). If compression is supported, records may be selectively compressed by changing the value of the COMPRESSION attribute.

FILESTATE

The FILESTATE attribute has the following values:

0 = CLOSED

The initial state of a subfile is CLOSED. The subfile returns to this state when it is closed by the user.

1 = AWAITINGHOST

This state indicates that the host specified by the HOSTNAME subfile attribute is unreachable. The subfile will remain in this state until the host becomes reachable. The FILESTATE may then change to OFFERED, OPENED, or CLOSED. I/O operations are not valid when the file is in this state.

2 = OFFERED

A subfile enters this state when an open has been done and the host specified by HOSTNAME is reachable, but no matching subfile has been found. I/O operations are not valid when the file is in this state.

3 = OPENED

This state indicates that the subfile is open and may be used to send or receive data.

4 = SHUTTINGDOWN

This state indicates that the system operator has requested that communications with the host involved in the subport dialog be terminated. This notification gives the program the opportunity to terminate in an orderly way. The port remains open and all I/O operations are valid.

5 = BLOCKED

This state indicates that the remote host has become temporarily unreachable. The port remains open and all I/O operations are valid.

6 = CLOSEPENDING

This state indicates that the user has closed the subfile, but the remote subfile has not yet acknowledged the closure. When close acknowledgment is received, FILESTATE changes to CLOSED.

7 = DEACTIVATIONPENDING

This state indicates that the other subfile has been closed and that this subfile has data queued for input.

8 = DEACTIVATED
This state indicates that the other subfile has been closed and that this subfile does not have data queued for input. Close is the only valid operation for a subfile in this state.

FRAMESIZE
This attribute has the same semantics that it has for other types of files. Data is always transmitted in 8-bit units, but the user program may deal with the data using other values for FRAMESIZE.

HOSTNAME
The HOSTNAME attribute specifies the host in the network which contains the remote process with which this local process wishes to communicate using this local subfile.

INPUTEVENT
If accessed as a file attribute, INPUTEVENT returns "happened" if the CENSUS file attribute is greater than zero. If accessed as a subfile attribute, INPUTEVENT returns "happened" if the CENSUS subfile attribute is greater than zero for the specified subfile.

INTNAME
The INTNAME of a port variable is the name by which it is declared and referenced by the process which is using it.

LASTSUBFILE
This attribute contains the subfile index of the last subfile that was used in an I/O operation on the file. This value is updated only if the I/O operation is successful.

MAXCENSUS
MAXCENSUS specifies the number of input messages that can be queued for this subfile before the other subfile is given a "NO BUFFER AVAILABLE" indication.

MAXRECSIZE
The MAXRECSIZE attribute can be accessed or assigned as a file attribute and is access-only as a subfile attribute. As a file attribute, MAXRECSIZE is used to access or assign the maximum message text size for the port file. When interrogated as a subfile attribute, MAXRECSIZE returns the actual message text size for the subfile. The value of the subfile MAXRECSIZE is negotiated when the subfile is being opened and may be different for each subfile.

MAXSUBFILES
This file attribute defines the maximum number of subfiles that can be opened for the file. The subfiles are assigned indices from 1 to MAXSUBFILES, inclusive.

MY HOST NAME

The MY HOST NAME attribute contains the name of the Host at which the process using the port is executing. It is used by the Port Level Manager in opening subport dialogs.

MY NAME

The MY NAME attribute is a string of up to 100 characters used to identify a port during the subfile matching process.

OUTPUTEVENT

OUTPUTEVENT is caused whenever output buffers become available and is reset by the system whenever no output buffers are available.

SECURITYGUARD

The SECURITYGUARD attribute of a file is the name of the guard file. Refer to Section 7, Access Control.

SECURITYTYPE

The SECURITYTYPE attribute of a file determines the class of users who may access the file. Refer to Section 7, Access Control.

STATE

The STATE attribute returns result information about the last I/O that was done on the file. The following STATE bits apply to port files:

STATE Bit 1 (high order bit)

This bit indicates that an error has occurred and is set in conjunction with other STATE bits.

STATE Bit 3

This bit indicates that an invalid subfile index was specified for an I/O operation.

STATE Bit 8

This bit indicates that an I/O operation failed for one of the following reasons:

a) A broadcast write failed for at least one subfile.

b) A write with the DONTWAIT option was not done because no buffer was available.

c) A read with the DONTWAIT option was not done because no data was available.

STATE Bit 9

This bit indicates end-of-file.

SUBFILEERROR

The SUBFILEERROR attribute is set to one of the following values after each I/O, OPEN, or CLOSE operation that affects the subfile:

0 = NOERROR
No error occurred during the subfile operation.

1 = DISCONNECTED
Communication with the other subfile has been severed due to network failure.

2 = DATALOST
During a close operation, all data was not transmitted successfully to the other subfile before the subfile was closed.

3 = NOBUFFER
An attempted write to this subfile failed because no buffer space was available. This error can occur only if DONTWAIT was specified on the write.

4 = NOFILEFOUND
An attempted OPEN on this subfile resulted in a NOFILEFOUND result.

5 = UNREACHABLEHOST
An attempted OPEN on this subfile resulted in an UNREACHABLEHOST result.

TITLE

The TITLE of a file is the name by which port files in communication with each other are associated. It is a single name of up to 17 characters and must not be null.

YOURNAME

The YOURNAME of a subfile is the same name as the MYNAME of the remote file with which it is communicating. It is a string of up to 100 characters.

YOURUSERCODE

The default value for the YOURUSERCODE attribute is the usercode of the task opening the subfile. Setting the value of YOURUSERCODE to null sets the value back to the default.

OPEN OPERATIONS

An OPEN operation on a subfile causes the system to find a remote endpoint that matches the subfile and to set up a dialog with that matching subfile.

A subfile provides a two-way, point-to-point logical communication path between two programs. In order to establish this path, each program must describe the desired connection. The system compares connection descriptions, matches complementary descriptions, and marks the subfiles OPENED. This process is called the matching algorithm. The following attributes are used by the matching algorithm:

HOSTNAME
: HOSTNAME contains the name of the host on which the complementary program is running. The value of HOSTNAME for each subfile must match the value of MYHOSTNAME for the complementary subfile. A null value for HOSTNAME matches any MYHOSTNAME value.

MYNAME
: In order to match, the value of MYNAME must match the value of YOURNAME for the complementary subfile. A null value for MYNAME matches only a null value for YOURNAME.

YOURNAME
: In order to match, the value of YOURNAME must match the value of MYNAME for the complementary subfile. A null value of YOURNAME matches any value for MYNAME.

TITLE
: The TITLE must match the TITLE of the complementary file. The default TITLE is the value of the INTNAME attribute.

SECURITYTYPE
: Each host performs security checking for its own subfile. If the value of its SECURITYTYPE attribute is PUBLIC, security checking is immediately successful. If the value is PRIVATE, the value of its YOURUSERCODE attribute must match the usercode of the process offering the complementary subfile. If the value is GUARDED, the result of the security check depends on the contents of the file named by the SECURITYGUARD attribute.

The OPEN statement requires two parameters: the subfile that is to be opened and an open option. The processing of the OPEN statement causes the matching algorithm to be invoked and causes a result to be returned indicating the success or failure of the open attempt. Acceptable values for the two parameters and the possible values for the OPEN statement result are described in the following paragraphs.

The subfile to be opened is specified by a subfile index. If the subfile index is zero, all subfiles with a FILESTATE of CLOSED are opened. When this "open all" facility is used, open results can be obtained by interrogating the SUBFILEERROR attribute. If the subfile index is greater than zero but not greater than MAXSUBFILES, the specified subfile is opened. If no subfile index is specified and MAXSUBFILES is greater than one, an error of BADSUBFILEINDEX is returned (the open results are described below); if MAXSUBFILES is equal to one, the (only) subfile will be opened.

The following three OPEN options apply to port files:

WAIT
: WAIT is the default value. The subfile is offered for matching, and the program is suspended on a "NO MATCHING PORT" condition until a matching subfile is found. If the host specified by HOSTNAME is UNREACHABLE, the program is suspended on a "NO MATCHING PORT" condition. If the host specified by HOSTNAME becomes unreachable before the open is complete, the program is resumed and UNREACHABLEHOST is returned.

OFFER
: OFFER causes the subfile to be offered for matching, and the program is resumed without waiting for the subfile to be matched.

AVAILABLE
: AVAILABLE causes the subfile to be matched only to a complementary subfile that has been already offered. If a match is found, the subfile is opened. If no match is found, a NOFILEFOUND result is returned (the subfile is NOT left offered for subsequent matching).

If WAIT or OFFER is specified as the open option and the subfile HOSTNAME names an unknown or unreachable host, the FILESTATE will be set to AWAITINGHOST. The subfile will remain in this state until either it is closed by the user or the host becomes reachable (the open then proceeds normally). If open AVAILABLE is specified and the host is unknown or unreachable, an open result of UNREACHABLEHOST is returned.

The OPEN function may return the following values as its result:

1 = OK
The open was successful. If the open type was OFFER, this result indicates that the open process was successfully started.

2 = NOFILEFOUND
NOFILEFOUND is returned if AVAILABLE was specified for the open option and a matching subfile was not found.

38 = UNREACHABLEHOST
UNREACHABLEHOST is returned in the case where open AVAILABLE was specified and the HOSTNAME attribute names an unreachable or unknown host. UNREACHABLEHOST is also returned in the case where open WAIT was specified and the host named by the HOSTNAME attribute becomes unreachable during the process of opening the subfile.

40 = ALREADYOPEN
ALREADYOPEN is returned if the specified subfile does not have FILESTATE equal to CLOSED.

42 = BADSUBFILEINDEX
BADSUBFILEINDEX is returned if the subfile index specified was less than zero or greater than MAXSUBFILES.

44 = OPENALLERROR
OPENALLERROR is returned if an open error occurs on any subfile when attempting to open all subfiles.

As part of the OPEN process, the value of the MAXRECSIZE attribute to be used in the conversation between the two subfiles is negotiated. The negotiated value is always the smaller of the two MAXRECSIZE values.

For languages that allow implicit file open, port files may be implicitly opened by an I/O operation on the file. If a non-zero subfile index is specified, then only that subfile will be opened. If a subfile index of zero is specified, then no implicit open action will take place. If a subfile index is not specified and MAXSUBFILES is equal to one, then the subfile will be implicitly opened.

I/O OPERATIONS

A program can perform a Read or Write to a subfile. Each subfile has a unique subfile index. By specifying a subfile index, the program can select a particular subfile. If the subfile index is omitted and MAXSUBFILES is equal to one, the I/O is performed on the only subfile of the file. If a subfile index of zero is specified on a read, a non-selective read is performed. The non-selective read gives the user the ability to read the next message from any subfile. If a subfile index of zero is specified for a write, the message is broadcast to all open subfiles. If an error occurs on any subfile during a broadcast write, the result from the write will indicate an error has occurred. The user can test the SUBFILEERROR attribute to determine the cause of the error.

If a user attempts an I/O that is larger than MAXRECSIZE, the message is truncated to a length of MAXRECSIZE. No indication of this truncation is returned to the user. The user can detect whether or not truncation has taken place by comparing the length of the attempted I/O with the MAXRECSIZE of the subfile.

I/O statements include a DONTWAIT option, which allows the program to continue if the I/O operation cannot be completed immediately. If an I/O statement is prematurely terminated because of a DONTWAIT specification, a field in the STATE attribute is set to indicate this occurrence. If a READ statement is executed and CENSUS is equal to zero, the program is suspended until data is available unless DONTWAIT is specified. If a WRITE statement is executed and no buffers are available, the program is suspended until the write can be completed unless DONTWAIT is specified.

If a subfile index less than zero or greater than MAXSUBFILES is specified or if no subfile index is specified and MAXSUBFILES is greater than one, the program attempting the I/O operation is terminated.

When the value of the FILESTATE attribute becomes DEACTIVATIONPENDING, all subsequent write operations return an end-of-file indication. Read operations will continue to operate normally as long as there are messages queued for input. When there are no more messages queued for input, the FILESTATE changes from DEACTIVATIONPENDING to DEACTIVATED, and all subsequent read operations return an end-of-file indication. If a program is suspended waiting for an I/O operation to complete and the remote subport closes (FILESTATE goes to DEACTIVATED), the program is resumed and end-of-file is returned.

I/O statement options such as skip-to-channel, skip-lines, stacker, timelimit, etc., are ignored for port files.

When the value of the FILESTATE attribute becomes BLOCKED, I/O operations continue to function normally. This action makes it possible for programs to attempt to maintain dialogs through temporary host-unreachable conditions. However, the system may quickly run out of buffer space and write operations will cause the program to be suspended unless DONTWAIT is specified. Since

there will be no incoming messages during this time, read operations will cause the program to be suspended once the queued input is exhausted, unless DONTWAIT is specified.

When the value of the FILESTATE attribute is OFFERED or AWAITINGHOST, all I/O operations will return an end-of-file condition.

The length of data transfers through a port file is dependent on the following criteria:

a) The length indicated in the I/O statement.

b) The MAXRECSIZE of the subfile.

c) For READ operations, the length of the data actually received (this is a factor only if BLOCKSTRUCTURE is equal to EXTERNAL).

d) The size of the user's buffer.

For WRITE operations, the amount of data sent is the minimum of the subfile MAXRECSIZE, the length indicated in the WRITE statement, and the size of the user's buffer. If an attempt is made to write data that is larger than MAXRECSIZE, the message is truncated to MAXRECSIZE. No indication of this truncation is given to the user. If the length specified in the WRITE is smaller than MAXRECSIZE but larger than the user's buffer, the message is truncated to the size of the user's buffer. No indication of this truncation is given to the user. The value of the BLOCKSTRUCTURE attribute has no effect on WRITE operations.

For READ operations with BLOCKSTRUCTURE equal to FIXED, the length of data delivered to the user's buffer is the minimum of the length indicated in the Read statement, the value of MAXRECSIZE, or the size of the user's buffer. If the message received is smaller than this size, the buffer is blank-filled. If the message received is larger than this size, the message is truncated without any indication given to the user. The value of CURRENTRECORD is always equal to MAXRECSIZE in this case.

For READ Operations with BLOCKSTRUCTURE equal to EXTERNAL, the length of the data delivered to the user's buffer is the minimum of the data indicated in the READ statement, the value of MAXRECSIZE, the size of the user's buffer, or the size of the actual message. If the size of the actual message is larger than the length indicated in the READ statement, the value of MAXRECSIZE, or the size of the user's buffer, truncation will take place and the user will not be notified. No blank fill is done for BLOCKSTRUCTURE equal to EXTERNAL. The CURRENTRECORD attribute may be used to determine the length of actual message.

CLOSE OPERATIONS

The Close operation terminates the dialog with the remote subfile and returns the system resources used by the local subfile. The remote subfile becomes DEACTIVATED.

The execution of a CLOSE statement changes the FILESTATE of the specified subfile to CLOSEPENDING or CLOSED. Any messages that have been queued for receipt are discarded.

Closing a port file may take a significant amount of time. Because this delay can be unacceptable for time-critical programs, the close option DONTWAIT is provided. For all close options except DONTWAIT, the program will be suspended while the actual close takes place. If DONTWAIT is specified, control returns immediately to the program and the process of actually closing the port takes place in parallel with the execution of the program. The program may detect when the close is complete by monitoring state changes. All other close options (PURGE, LOCK, etc.) are ignored. The CLOSE function may return the following values as its result:

1 = OK
The close was successful.

30 = FILENOTOPEN
The subfile was already closed.

32 = DATALOST
All data was not successfully delivered to the destination subfile.

42 = BADSUBFILEINDEX
The subfile index specified was less than zero or greater than MAXSUBFILES.

46 = CLOSEALLERROR
CLOSEALLERROR is returned if a close error occurs on any subfile when attempting to close all subfiles.

If the subfile index specified in a CLOSE statement is zero, all open subfiles are closed. When this "close all" facility is used, close results may be obtained by interrogating the SUBFILEERROR attribute. If the subfile index is greater than zero but not greater than MAXSUBFILES, only the specified subfile is closed. If no subfile index is specified and MAXSUBFILES is greater than one, an error of BADSUBFILEINDEX is returned. If MAXSUBFILES is equal to one, the (only) subfile will be closed.

SAMPLE PROGRAMS

The sample "COBOL74 PROGRAM TO DO A SIMPLE READ AND WRITE TO A PORT FILE" program contains examples of the use of the Open, Close, Read, and Write statements. The complementary program, which is executed in a remote host, is identical except that the functions are reversed.

Line 10600 selects the filename KEN and associates it with a PORT file. Line 11100 defines file KEN with a field of 8 alphanumeric characters called PORT-REC. Line 11600 sets the subfile attribute HOSTNAME (the name of the host on which the complementary program is running) to HOSTA. Line 11700 opens the file KEN for both input and output. No subfile index is specified, but that is valid since there is only one subfile. No open option is specified, the default value of WAIT will be used. The only subfile will be offered for matching and the program will be suspended until the matching subfile is found. Line 11800 writes the message "MESSAGE1" to the subfile. The complementary program in HOSTA will read it and send back "MESSAGE2". Line 12000 reads the message "MESSAGE2". DONTWAIT has not been specified, so the program is suspended until data is available (i.e., CENSUS is not equal to zero). Line 12300 closes the file. Since DONTWAIT is not specified, the program will be suspended again while the actual close takes place.

```
10000*COBOL74 PROGRAM TO DO A SIMPLE READ AND WRITE TO A PORT FILE.

10100*THIS PROGRAM IS TO BE EXECUTED ON HOST "HOSTB"
10200 IDENTIFICATION DIVISION.
10210 PROGRAM I-D. READ-WRITE-TEST-B.
10300 ENVIRONMENT DIVISION.
10320  CONFIGURATION SECTION.
10330   SOURCE-COMPUTER. B-6800.
10340   OBJECT-COMPUTER. B-6800.
10400 INPUT-OUTPUT SECTION.
10500 FILE-CONTROL.
10600     SELECT KEN ASSIGN TO PORT.
10800 DATA DIVISION.
10900 FILE SECTION.
11000 FD KEN.
11100 01 PORT-REC                PIC X(8).
11300 PROCEDURE DIVISION.
11500 SAMPLE-PROGRAM.
11600       CHANGE ATTRIBUTE HOSTNAME OF KEN TO "HOSTA".
11700       OPEN I-O KEN.
11800       MOVE "MESSAGE1" TO PORT-REC.
11900       WRITE PORT-REC.
12000       READ KEN AT END DISPLAY "ERROR ON READ".
12100       IF PORT-REC IS NOT EQUAL TO "MESSAGE2"
12200         DISPLAY "ERROR".
12300       CLOSE KEN.
12400       STOP RUN.
```

```
20000*COMPLEMENTARY COBOL74 PROGRAM FOR READ/WRITE TO A PORT FILE

20100*THIS PROGRAM IS TO BE EXECUTED ON HOST "HOSTA"
20200 IDENTIFICATION DIVISION.
20250 PROGRAM I-D. READ-WRITE-TEST-A.
20300 ENVIRONMENT DIVISION.
20310  CONFIGURATION SECTION.
20320   SOURCE-COMPUTER. B-1900.
20330   OBJECT-COMPUTER. B-1900.
20400 INPUT-OUTPUT SECTION.
20500 FILE-CONTROL.
20600     SELECT KEN ASSIGN TO PORT.
20800 DATA DIVISION.
20900 FILE SECTION.
21000 FD KEN.
21100 01 PORT-REC                  PIC X(8).
21300 PROCEDURE DIVISION.
21500 SAMPLE-PROGRAM.
21600       CHANGE ATTRIBUTE HOSTNAME OF KEN TO "HOSTB".
21700       OPEN I-O KEN.
21800       READ KEN AT END DISPLAY "ERROR ON READ".
21900       IF PORT-REC IS NOT EQUAL TO "MESSAGE1"
22000         DISPLAY "ERROR".
22100       MOVE "MESSAGE2" TO PORT-REC.
22200       WRITE PORT-REC.
22300       CLOSE KEN.
22400       STOP RUN.
```

The sample "COBOL74 READ/WRITE PROGRAM USING BNA OPTIONS" program declares a PORT file with 5 subfiles. It opens all the subfiles with the OFFER open option, and the program is resumed without waiting for the subfiles to be matched. It waits for the CHANGEEVENTs to determine that at least 3 subfiles have opened, then broadcasts a message "SEND ME A MESSAGE" to all of the open subfiles. The program does a non-selective read, and is suspended until an input message is available. It determines which process sent the message from the content of the ACTUAL KEY associated with the PORT, then does a selective write to the subfile associated with that process, sending the message "YOU WIN". Then it closes that subfile, and does a broadcast write to all remaining open subfiles, sending the message "YOU LOSE". Finally, it closes all the files.

```
30000* COBOL74 READ/WRITE PROGRAM USING BNA OPTIONS.
30100*
30200 IDENTIFICATION DIVISION.
30300 PROGRAM-ID. TEST-PROGRAM.
30400 ENVIRONMENT DIVISION.
30500 CONFIGURATION SECTION.
30600   SOURCE-COMPUTER. B-6800.
30700   OBJECT-COMPUTER. B-6800.
30800 INPUT-OUTPUT SECTION.
30900 FILE-CONTROL.
31000     SELECT PATH ASSIGN TO PORT
31100            ACTUAL KEY IS PATH-AK
31200            FILE STATUS IS PATH-STATUS.
31300*
31400 DATA DIVISION.
31500 FILE SECTION.
31600 FD PATH.
31700 01 PORT-REC            PIC X(18).
31800 WORKING-STORAGE SECTION.
31900 01 PORT-WS.
32000    05 PATH-STATUS      PIC XX.
32100    05 PATH-AK          PIC 9 COMPUTATIONAL.
32200    05 NUMBER-OPENED    PIC 9 COMPUTATIONAL.
32300*
32400 PROCEDURE DIVISION.
32500*
32600 DECLARATIVES.
32700 IO-ERROR SECTION.
32800 USE AFTER ERROR PROCEDURE ON I-O.
32900 IO-ERROR-HANDLER.
33000      DISPLAY "I/O ERROR ENCOUNTERED".
33100      DISPLAY "STATUS IS ",PATH-STATUS.
33200 END DECLARATIVES.
```

```
33300*
33400 THE-PROGRAM SECTION.
33500 INITIALIZATION.
33600       CHANGE ATTRIBUTE MAXSUBFILES OF PATH TO 5.
33700       CHANGE ATTRIBUTE MYNAME OF PATH TO "MASTER".
33800       CHANGE ATTRIBUTE YOURNAME OF PATH(0) TO "COMPANION".
33900       CHANGE ATTRIBUTE HOSTNAME OF PATH(0) TO "HOSTA".
34000       MOVE 0 TO NUMBER-OPENED.
34100 OPEN-THE-PORT.
34200       MOVE 0 TO PATH-AK.
34300       OPEN OFFER PATH.
34400 WAIT-FOR-THREE-TO-OPEN.
34500       WAIT ATTRIBUTE CHANGEEVENT OF PATH.
34600       ADD 1 TO NUMBER-OPENED.
34700       IF NUMBER-OPENED IS LESS THAN 3
34800         GO TO WAIT-FOR-THREE-TO-OPEN.
34900* FOR SIMPLICITY THE ABOVE PARAGRAPH IGNORES SOME POTENTIAL
35000* EXCEPTION CONDITIONS. A MORE THOROUGH PROGRAM COULD KEEP
35100* A TABLE OF PREVIOUS FILESTATES AND CHECK THE FILESTATE
35200* ATTRIBUTE OF THE CHANGEDSUBFILE TO INSURE THAT AT
35300* LEAST 3 ARE OPEN.
35400 BROADCAST-THE-MESSAGE.
35500       MOVE 0 TO PATH-AK.
35600       MOVE "SEND ME A MESSAGE" TO PORT-REC.
35700       WRITE PORT-REC.
35800 GET-A-MESSAGE.
35900       MOVE 0 TO PATH-AK.
36000       READ PATH.
36100 SEND-A-MESSAGE.
36200* THE ACTUAL KEY AT THIS POINT CONTAINS THE SUBFILE
36300* FROM WHICH THE MESSAGE WAS REMOVED IN THE LAST
36400* READ STATEMENT.
36500       MOVE "YOU WIN" TO PORT-REC.
36600       WRITE PORT-REC.
36700 CLOSE-THE-WINNING-SUBFILE.
36800       CLOSE PATH.
36900 BROADCAST-A-MESSAGE-TO-LOSERS.
37000       MOVE "YOU LOSE" TO PORT-REC.
37100       MOVE 0 TO PATH-AK.
37200       WRITE PORT-REC.
37300 CLOSE-THE-LOSERS.
37400       CLOSE PATH.
37500       STOP RUN.
```

SECTION 4

FUNCTIONAL DESCRIPTION - HOST SERVICES

GENERAL

Host Services provides the functions associated with distributed processing in a manner such that distributed processing appears to the user to be almost as simple as accessing the resources of a single system. This is done through extensions to features already available to the user such as file attributes, task attributes, library maintenance, and WORK FLOW LANGUAGE (WFL).

Host Services uses Network Services to transport its data between hosts in the network. Section 3 of this manual describes the use of Network Services by user programs for communication with remote user programs. A Host Services process communicates with a remote Host Services process in the same way, through Network Services ports.

Host Services is implemented as a set of host-to-host protocols, one protocol for each Host Services function. A protocol defines a group of messages designed to support a particular function in terms of the intent and format of the messages, and the allowable sequence of the messages. The protocols are:

1) the Operator Display Terminal (ODT) protocol, which manages the transfer of operator terminal input and related output messages between hosts.

2) the Job Transfer protocol, which provides the ability to transfer a series of job source images from a local host to a remote host for interpretation and processing.

3) the Logical I/O protocol, which allows tasks to create, read from, and update files on remote hosts.

4) the Remote Tasking protocol, which supports programmatic task initiation, monitoring, and control on a remote host.

5) the Station Transfer protocol, which allows a terminal which is physically attached to one host to appear as if it were attached to another.

6) the Status Change protocol, which is used to report changes in the status of tasks which were initiated by one host but are running on another.

Host Services supports an additional function, File Transfer. File Transfer is used to copy complete files from a remote host to the local host, from the local host to a remote host, or from a remote host to another remote host. File Transfer is provided by a utility, rather than a protocol. The File Transfer utility uses the Logical I/O protocol to copy complete files to/from a remote host.

DIALOGS AND PROTOCOLS

A "protocol" is the set of all valid message (request/response) sequences between two Host Services processes. Within Host Services, the term "dialog" is used to describe the specific series of messages transmitted between two Host Services processes in order to accomplish a particular function. For example, the Job Transfer protocol describes all valid messages between a Host Services process on one host which is transferring a job to a Host Services process on another host. A Job Transfer dialog is a particular instance of this protocol, using whichever messages are appropriate.

The set of protocols supported by a particular host is host-dependent. A host responds with an error reponse if it receives a dialog initiate request for a protocol which it does not support.

Every dialog goes through three phases: initiation, operation and termination. In the initiation phase, the two hosts establish communication and specify the parameters of the dialog. The initiating host starts the dialog by opening a port to communicate with the remote host. Host Services ports have a special usercode, the 24 bit hexadecimal string "BADBAD". Network Services at the remote node alerts its Host Services when it detects an OFFER with that special usercode. Host Services at the remote host then opens a port to match the offered port. A user of Host Services can therefore access the resources of a remote host without being required to establish a complementary user process at the remote host.

Most of the actual work performed in the dialog is done in the operation phase. The operation phase performs the necessary functions for one logical operation (e.g. in the Logical I/O protocol, the opening of a file, transfer of data to/from the file, and closing of the file. In the operation phase, the flow of messages is either basically uni-directional or basically bi-directional, depending on the protocol. In a uni-directional dialog, such as a Job Transfer dialog, one host has control of the dialog. Since the host in control sends most of the requests, the flow of information is largely in one direction. A bi-directional dialog, such as an ODT dialog, is designed for the simultaneous flow of requests in both directions.

The termination stage terminates the dialog and may be invoked by either host.

HOST SERVICES PROTOCOLS

OPERATOR DISPLAY TERMINAL (ODT)

The Operator Display Terminal (ODT) protocol is used to transfer operator input messages entered by the system operator at the local host to a remote host. The ODT protocol also transfers the remote host's responses to those messages back to the local host.

Depending on the capabilities provided by the local host, users can also have access to this interface. For example, a time-sharing system might allow a user to enter system messages through a terminal.

Syntax

The following syntax is used by an operator (or user) to communicate directly with a remote host:

```
-- AT --<hostname>--<request>----------|
```

The word "AT" indicates that the destination for the <request> is a remote host, designated by <hostname>. The <request> is text that the remote host is to interpret as an operator input message. The operator input is not checked at the local host for valid syntax. All input is checked at the remote host. For example, the following ODT input requests the time at the remote host "NORTHPOLE":

```
AT NORTHPOLE WT
```

When ODT output is received from the remote host, the local host prefaces the text with the <hostname> of the host that sent the ODT output. An output similar to the following could be displayed in response to the above request:

```
***** FROM NORTHPOLE *****
DATE IS THURSDAY DEC 25, 1980 (80360) 02:30:07
```

Chaining of commands is not allowed. For example, "AT NORTHPOLE AT MIAMI WT" is not a valid command.

Access Control

Since the local host does not interpret the operator input, it does not know what user identification information is required. Thus, all available identification (usercode, accesscode, and/or chargecode) is sent with each request. The identification sent can be that associated with the system console, that associated with the user making the request (e.g., through an MCS), or the default identification for the system. The remote host uses the hostname/usercode combination to determine whether the request is allowable for that user. The accesscode and chargecode may be required by the remote host for certain requests.

Dialog

The ODT protocol is bi-directional, that is, neither host has primary control. Only one ODT dialog may be in progress between each pair of hosts. The dialog is started when an operator enters an ODT command for a remote host, or it may be started by other protocols. The dialog is terminated when the host pair has completed communications or when a timeout occurs.

Port Description

The port attributes for the ODT port are as follows:

```
MYNAME       = "ODTDIALOG"
YOURNAME     = "ODTDIALOG"
PORTNAME     = "ODTPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

JOB TRANSFER

The Job Transfer protocol provides the ability to transfer job files from the local (sending) host to a remote (receiving) host for interpretation and processing on the remote host. Both job source images (data records) and control records can be transferred. Binary decks are not supported by this protocol.

Syntax Examples

The following syntax can be used to transfer a job file to a remote host:

```
?AT ----<hostname>--<job>--------------------|
```

The word "AT" indicates that the destination for the <job> is a remote host, designated by <hostname>. The <job> is interpreted by the remote host for execution at the remote host. The records transferred are not checked at the local (sending) host for valid syntax. Job Transfer to a remote host may also be initiated using the START command from a user terminal, from another job, or from an ODT. The format is:

```
?START <file name>; HOSTNAME = <host name>
```

The file referenced by <file name> is a file of job source records in the same format as the <job> records in the example above. Refer to documentation for the type of system at the remote host for details of syntax. The following are examples of the statements used for Job Transfer.

```
Example 1 -
          ?AT <HOSTNAME> STREAM <NAME>
          <JOB IMAGE>
          <JOB IMAGE>
               .
               .
          <JOB IMAGE>
          ?TERMINATE <NAME>


Example 2 -
          ?START JOB <FILE NAME>; HOSTNAME = <HOSTNAME>

Example 3 -
          ?AT <HOSTNAME> BEGIN JOB
          <JOB IMAGE>
          <JOB IMAGE>
               .
               .
          <JOB IMAGE>
          ?END JOB

Example 4 -
          ?START JOB <FILE NAME>
```

A record recognized by the local (sending) host as a control record (for example, one with an invalid character in column one as read through a card reader) is sent with the "control record" flag set to TRUE. The remote (receiving) host ignores the first character of the record and interprets the remainder of the data area of the record as control information. The sending host is responsible for detecting the end of the job deck. The receiving host uses the indication sent by the sender to infer the appropriate terminator record (unless a terminator record appeared just before the indication, in which case it need not be inferred).

Access Control

If there is a usercode associated with the job or operator performing the Job Transfer, that usercode is sent to the remote host. It and the name of the host from which the transfer originated make up the hostname/usercode pair used by the remote host. Also, the job deck itself can contain a usercode/password to be authenticated by the receiving host.

Dialog

The Job Transfer Protocol is a uni-directional protocol controlled by the initiating (sending) host. Jobs are transferred one at a time, but multiple jobs may be sent in sequence. Multiple Job Transfer dialogs between the same two hosts can be in progress at the same time. The sending host begins the dialog when it gets a Job Transfer request. Changes in status of a job or its offspring tasks are reported through the Status Change protocol. Operator input pertaining to the job is sent through the ODT protocol. The dialog is terminated when all of the job images have been transferred, even though the job may still be running or may not even have been started.

Port Descriptions

The port attributes for the sending host's Job Transfer port are as follows:

```
MYNAME       = "JOBXFERSEND"
YOURNAME     = "JOBXFERRECV"
PORTNAME     = "JOBXFERPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

The port attributes for the receiving host's Job Transfer port are as follows:

```
MYNAME       = "JOBXFERRECV"
YOURNAME     = "JOBXFERSEND"
PORTNAME     = "JOBXFERPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

LOGICAL I/O

Logical I/O allows programs to access files located at remote hosts in the same manner that they access files located at the local host. The statements required to access the file are the same as if the file were located in the local host, except that the program must indicate that the file is located on another host by setting the file's HOSTNAME attribute to the name of that host. Normal I/O operations are used by the program to create the file, read from it, and update it.

Syntax Examples

ALGOL EXAMPLE

```
BEGIN

    FILE SOURCE (KIND = DISK,UNITS = CHARACTERS, MAXRECSIZE = 80,
    FILENAME = "DATA/FILE",HOSTNAME = "HOST-B");


    FILE DEST (KIND = TAPE,UNITS - CHARACTERS, MAXRECSIZE = 80,
    BLOCKSIZE = 800, FILENAME - "DATA/FILE.", HOSTNAME =
    "HOST-A",NEWFILE - TRUE);


            EBCDIC ARRAY RECORD (0:79);
            LABEL EOF;

    WHILE TRUE DO

            BEGIN
            READ (SOURCE, 80, RECORD) [EOF];
            WRITE (DEST, 80, RECORD);
            END;

  EOF:

    CLOSE (SOURCE);
    LOCK (DEST);


END
```

COBOL EXAMPLE

```
IDENTIFICATION DIVISION.
    PROGRAM-ID. FILE-TRANSFER.


ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT REMOTE-FILE ASSIGN TO DISK.
    SELECT LOCAL-FILE ASSIGN TO DISK.


DATA DIVISION.
FILE SECTION.
FD REMOTE-FILE VALUE OF ID "REMFILE" HOSTNAME "HOST-B".
01 IN-RECORD           PIC X(80).


FD LOCAL-FILE VALUE OF ID "LOCFILE".
01 OUT-RECORD          PIC X(80).


PROCEDURE DIVISION.
MAIN SECTION.
P1.
    OPEN INPUT REMOTE-FILE.
    OPEN OUTPUT LOCAL-FILE.

P2.
    READ REMOTE-FILE ;AT END GO P4.
    MOVE IN-RECORD TO OUT-RECORD.
    WRITE OUT-RECORD ;INVALID KEY GO P3.
    GO P2.

P3.
    DISPLAY "ERROR IN WRITE".

P4.
    CLOSE REMOTE-FILE WITH LOCK.
    CLOSE LOCAL-FILE WITH LOCK.

P5.
    STOP RUN.
```

Access Control

The protocol requires that a usercode be associated with the task initiating the logical I/O operations, and this usercode becomes part of the operable hostname/usercode pair. If the value of SECURITYTYPE for the file is PRIVATE, access is granted only to a specified usercode. If it is GUARDED, access is granted only if the requestor satisfies the requirements of the guardfile.

In addition, a usercode can be part of the file title. It is a usercode at the host where the file resides and is used for file identification only.

Dialog

The Logical I/O protocol is a uni-directional protocol controlled by the "initiating" host. The initiating host is the one with the logical file. The host with the physical file is called the "cooperating" host. The initiating host begins the dialog when the user program opens the file.

The status of the Logical I/O operation is reported through the Status Change protocol. Responses to these reports are sent back with the ODT protocol.

The dialog is terminated when the user program closes the file. There may be as many Logical I/O dialogs per host as there are ports available. There is one Logical I/O dialog for each file.

In order to allow multiple buffering of data, the initiating host may have several data transfer requests outstanding. The cooperating host responds to these requests in the order in which they were received.

If any type of communication failure occurs while trying to access a file on a remote host, the file attribute STATE will have bits 8 and 0 set to TRUE to indicate that the error occurred.

Port Descriptions

The port attributes for the initiating host's Logical I/O port are as follows:

```
MYNAME       = "LIOINIT"
YOURNAME     = "LIOCOOP"
PORTNAME     = "LIOPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

The port attributes for the cooperating host's Logical I/O port are as follows:

```
MYNAME       = "LIOCOOP"
YOURNAME     = "LIOINIT"
PORTNAME     = "LIOPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

REMOTE TASKING

Some Burroughs products permit a job or task (process) to initiate, monitor, and control "sub-tasks". The Host Services Remote Tasking protocol allows these "sub-tasks" to be processed at a remote host.

The existing task attributes have been extended to include the HOSTNAME attribute. When the HOSTNAME attribute is set for a task, it indicates the name of the host where the program's code file is to be found and executed. Tasks on remote hosts are initiated in the same manner as tasks on the local host; by Work Flow Language (WFL) jobs, by application programs via a RUN, PROCESS, or CALL of an external code file, and by users of Command and Edit (CANDE) terminals. The program's code file must be resident at the host where it is to be executed. Once a task has been initiated at a remote host, the initiator has the same control capabilities that exist for tasks running on the local host; task attributes can be interrogated and set, and the task can be suspended, resumed and terminated.

Syntax Example (WFL)

```
RUN X ON P; HOSTNAME=B
```

This causes the codefile whose title is X ON P to be executed as a task on host B. The file X must reside on pack P of host B.

Access Control

The protocol requires that the parent job, task, or terminal be running under a usercode. This usercode and the name of the host at which the parent is running make up the hostname/usercode pair used by the receiving host.

Dialog

The Remote Tasking protocol is bi-directional. Only one Tasking dialog can be in progress between each pair of hosts. The initiating host begins the dialog when a parent task requests the initiation of a remote task. Once the dialog has been established, either host may initiate tasks at the other host by sending an appropriate request. The host actually running the task uses the Host Services Status Change protocol to report changes of status for the task. While the task is running, the host that requested its initiation can use the ODT protocol to send operator input pertaining to the remote task. At the conclusion of the task, the cooperating host sends the initiating host a list of attributes specifying the final state of the task, including necessary accounting data. The dialog is terminated when there are no more dependent tasks running on either host.

If the connection between the hosts is lost, any dependent tasks are terminated, to prevent uncontrolled continuation of dependent tasks and usage of resources by "hung" tasks waiting for communication with the remote parent task. Under normal circumstances, the dialog continues as long as either host is running a task for the other host. If the dialog must be ended abnormally, the tasks are treated as if communications had been lost.

Port Descriptions

The port attributes for the Remote Tasking port are as follows:

```
MYNAME       = "TASKDIALOG"
YOURNAME     = "TASKDIALOG"
PORTNAME     = "TASKPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

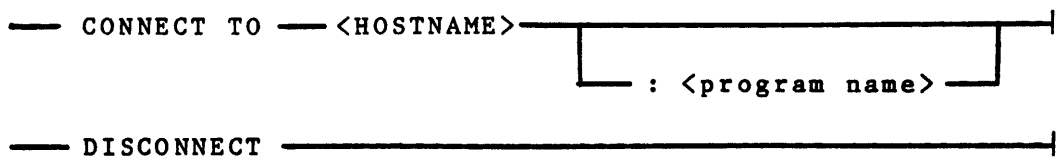
STATION TRANSFER

The Station Transfer protocol provides a means of logically attaching a terminal to a remote host. Physical control of the terminal is maintained by the host to which it is physically connected.

The terminal may be a part of an existing non-BNA network, or a part of a BNA network. The user of the terminal appears to have a direct connection to the remote host. If the remote host is not the same type as the local host, the user must operate in the environment of the remote host. That is, the syntax of the input and output messages are in the format used at the remote host.

Syntax Examples

The syntax to start and end the attachment of a terminal to a remote host is:



The <program name> is the file name of the program to which the station is to be connected.

Dialog

The dialog is started when the operator at the initiating host enters a CONNECT command (the CONNECT case), or when a process at the initiating host has a process to connect to a remote station (the ATTACH case). Once the cooperating host has agreed to begin the dialog, the initiating host sends either a STATION CONNECT request, if it has a station to connect to a remote process, or a STATION ATTACH request, if it has a process to attach to a remote station. The host that is connected to the station is called the "station host" (regardless of whether it is the initiating host or the cooperating host) and the host that is running the process is the "process host". The CONNECT and ATTACH cases are illustrated in Figures 4-1 and 4-2.

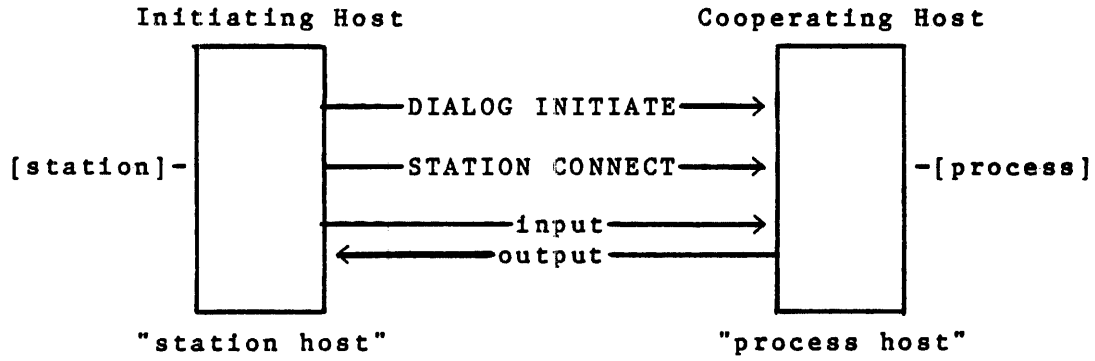


Figure 4-1. Station Transfer, The CONNECT Case

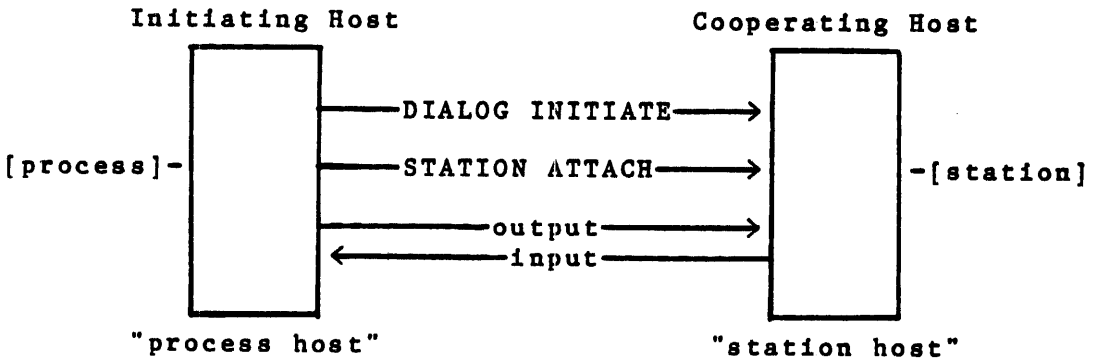


Figure 4-2. Station Transfer, The ATTACH Case

As part of the connect or attach procedure, the two hosts may negotiate the responsibility for handling certain exception conditions. The station host has the responsibility for all exceptions except those that the two hosts have agreed to transfer to the process host.

Once the logical connection has been made, any message from the station to the process or from the process to the station is transferred. In addition to the data, the messages may contain encoded information requesting that the receiving host perform certain higher-level data comm functions for this message, such as process acknowledgment ("logicalack") or, at the station end, cursor or carriage control. The process host may inquire about or modify the attributes of the station, and recall or purge output messages.

Either the station host or the process host can request that the station be logically disconnected from the process host.

The Station Transfer protocol is uni-directional, under the control of the initiating host. One dialog is initiated for every station/process pair. There can be as many Station Transfer dialogs per host as there are ports available.

Access Control

In the CONNECT case, where the initiating host has a terminal to connect to a remote process, the usercode of the user at the terminal is passed to the remote host. In the ATTACH case, where the initiating host has a process to attach to a remote station, the usercode of the process requesting the attachment is sent.

In addition, after the terminal is transferred, the user at the terminal must log on to the process host as though the terminal were directly connected to that host.

Port Descriptions

The port attributes for the initiating host's Station Transfer port are as follows:

```
MYNAME       = "STNINIT"
YOURNAME     = "STNCOOP"
PORTNAME     = "STNPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

The port attributes for the cooperating host's Station Transfer port are as follows:

```
MYNAME       = "STNCOOP"
YOURNAME     = "STNINIT"
PORTNAME     = "STNPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

STATUS CHANGE PROTOCOL

The Status Change protocol is used to report the status of all tasks initiated through any other protocol. Whenever such a task changes status categories or programmatically displays a message for the operator, the appropriate hosts are notified.

Syntax

There is no syntax associated with this protocol. It is a support protocol used by the other Host Services protocols, not by users or operators.

Access Control

No usercodes are required for this protocol. It is used only to support other, already authorized, protocols.

Dialog

The Status Change protocol is bi-directional. There is only one Status Change dialog in progress between each pair of hosts. The dialog is started when one of the other protocols requires it to send status change information. The dialog is terminated when the host pair has completed communications or when a timeout occurs.

Port Description

The port attributes for the Status Change port are as follows:

```
MYNAME       = "STATUSDIALOG"
YOURNAME     = "STATUSDIALOG"
PORTNAME     = "STATUSPORT"
SECURITYTYPE = PRIVATE
YOURUSERCODE = the 24 bit hexadecimal string "BADBAD".
```

HOST SERVICES UTILITIES

FILE TRANSFER

Using the File Transfer utility, an application program or operator can copy files from one host to another in the same way that files are copied from one device to another at a single host.

Files are copied from one host to another by using the HOSTNAME attribute as an extension to the library maintenance functions. An operator or user can copy files from the local host to a remote host by specifying a value for the destination HOSTNAME, from a remote host to the local host by specifying a value for the source HOSTNAME, or from a remote host to another remote host by specifying a value for both.

The File Transfer utility uses the Host Services Logical I/O protocol to achieve the reading and/or writing of a physical remote file.

Syntax Examples

```
COPY X FROM P (KIND=DISK) TO Q (KIND=DISK, HOSTNAME=B)
```

This causes file X ON P to be copied to Q at host B.

```
COPY A, B, X FROM OURPACK(KIND=DISK) TO
THEIRPACK(KIND=DISK, HOSTNAME=THEM)
```

This causes files A, B, and X to be copied from OURPACK at the local host to THEIRPACK at host THEM.

SECTION 5

FUNCTIONAL DESCRIPTION - NETWORK SERVICES

Network Services provides the interconnection and data transport mechanism for Host Services and for communication between user programs.

Within a node, Network Services is functionally divided into three levels and a Network Services Manager. Figure 5-1 shows the structure of a node.

The three levels, from lowest to highest, are: the Station Level, which provides physical link connections between neighbor nodes; the Router Level, which provides a logical connection from each node to every other node in the network; and the Port Level, which ensures reliable transfer of messages from senders to receivers. The Network Services Manager performs management functions to the three functional layers.

At the Station Level, one station exists for each physical link to a neighbor node. At the Port Level, one or more ports exist for each user program which utilizes remote access or communications. Each port has one subport for each remote user program with which it is communicating.

Each of these levels communicates with the corresponding level at other nodes through the lower levels. The BNA network can be viewed from the physical interconnection level (forming the STATION level network), from the logical interconnection level (forming the ROUTER level network) or from the user-to-user interface level (forming the PORT level network). These levels are described in this Section as the STATION LEVEL, ROUTER LEVEL, and PORT LEVEL.

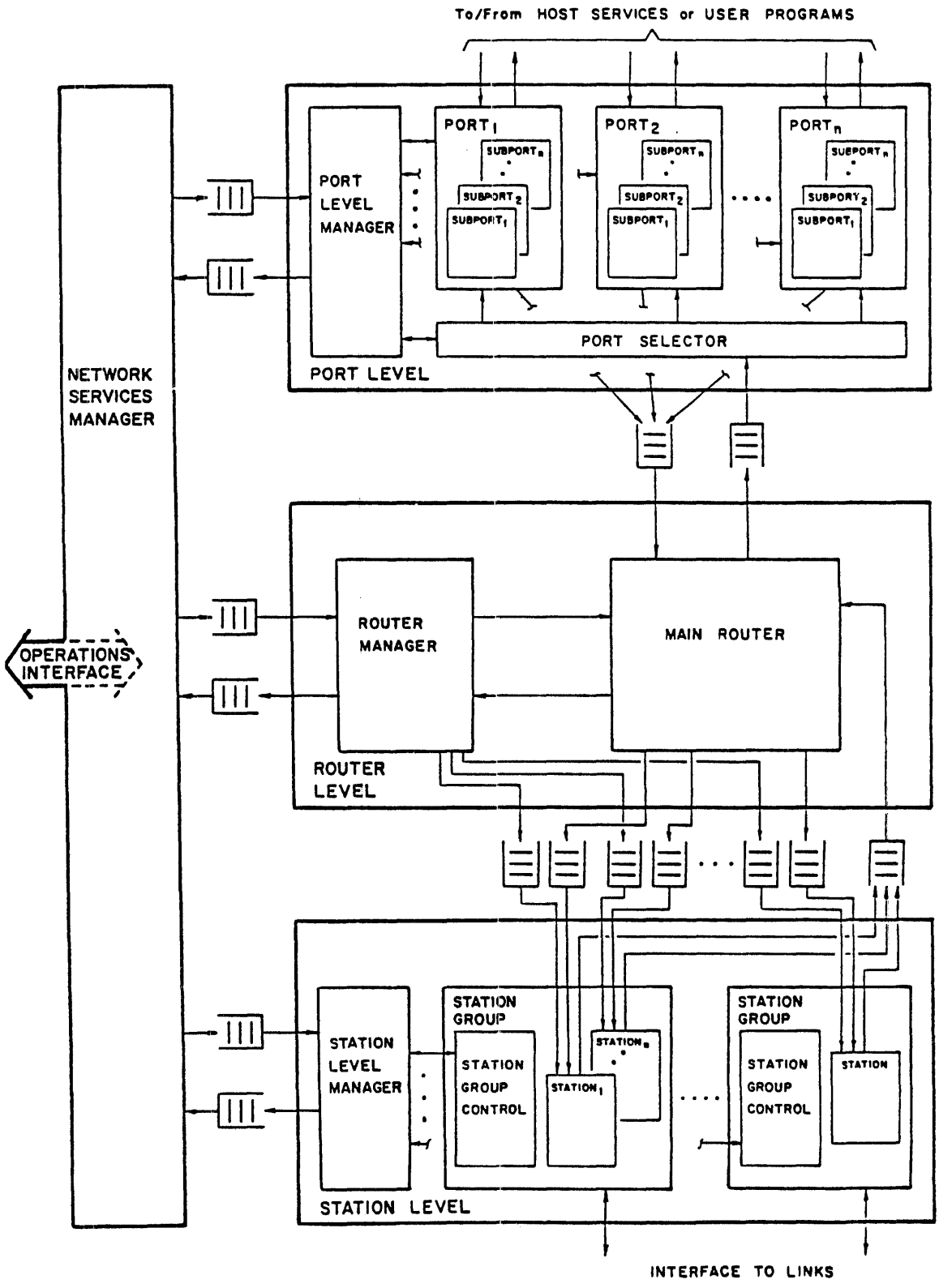


Figure 5-1. Network Services Block Diagram

PORT LEVEL

INTRODUCTION

The port level provides three major services:

- Communication between user programs
- Host Service communication
- Inter-host communication

Communication between user programs is a feature that allows user programs to send and receive messages. The feature provides the functions commonly referred to as Inter-Process Communication (IPC).

Host Services Communications are supported by the Port Level in two ways: (1) inter-process communication (IPC) provides the means for supporting Host Service dialogs, and (2) OFFERS for Host Service ports are handled uniquely. This subject is pursued later in this section as "HOST SERVICES AND THE PORTS.

Inter-host communications within a BNA network is used to establish and maintain the connections required for the above user and Host Services communications.

The manner in which the Port Level provides these services is described in this section. The functional components will first be described, followed by descriptions of major Port Level tasks, and finally the Port Level attributes. The descriptions of the attributes include their use and how they are set. For detail on any frame formats or frame related attributes mentioned in this section, refer to Section 9.

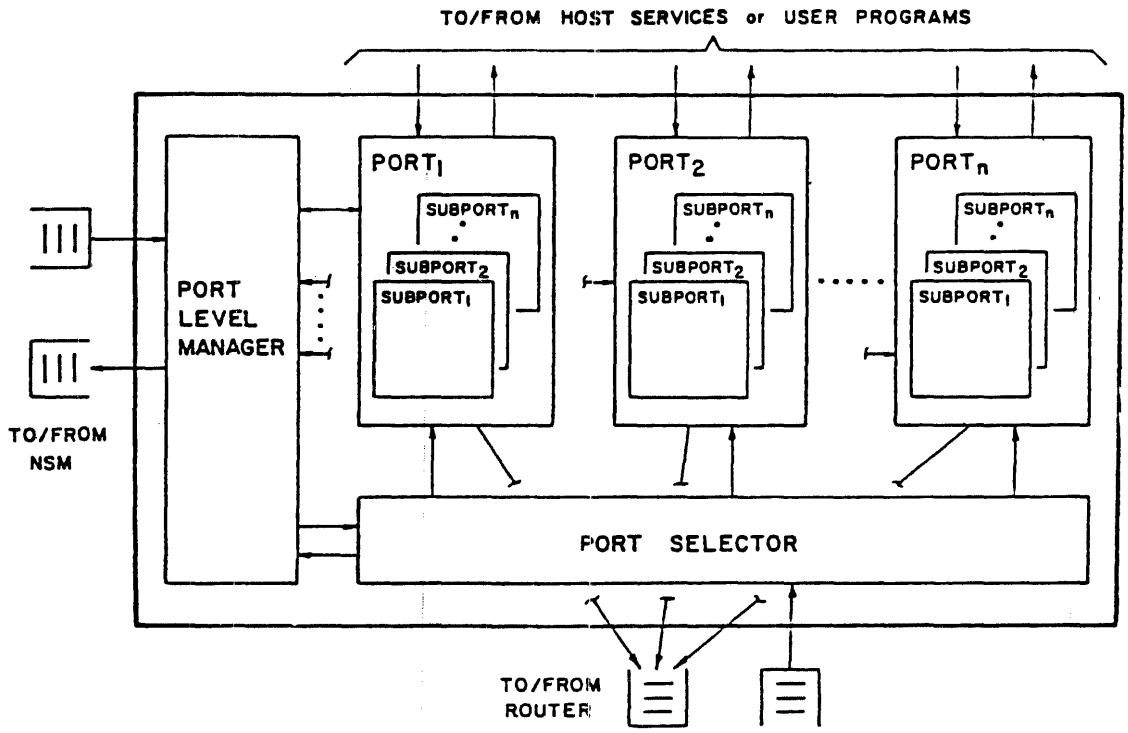


Figure 5-2. Port Level Interfaces

The Port Level consists of three major functional components: the Ports, the Port Selector, and the Port Level Manager (PLM), refer to Figure 5-2. These are discussed in the following paragraphs.

PORT SELECTOR

The function of the Port Selector is to receive port traffic from the Router and direct it to the required port. It checks that the header items and port addresses are present and valid, before directing them to their destination port.

PORT

The Port provides the end-point functions of a set of connection paths between two or more processes. Connection paths are always bi-directional. Messages can be sent and received simultaneously. Even when a user of a port uses a uni-directional flow of data, port control messages still occur in both directions.

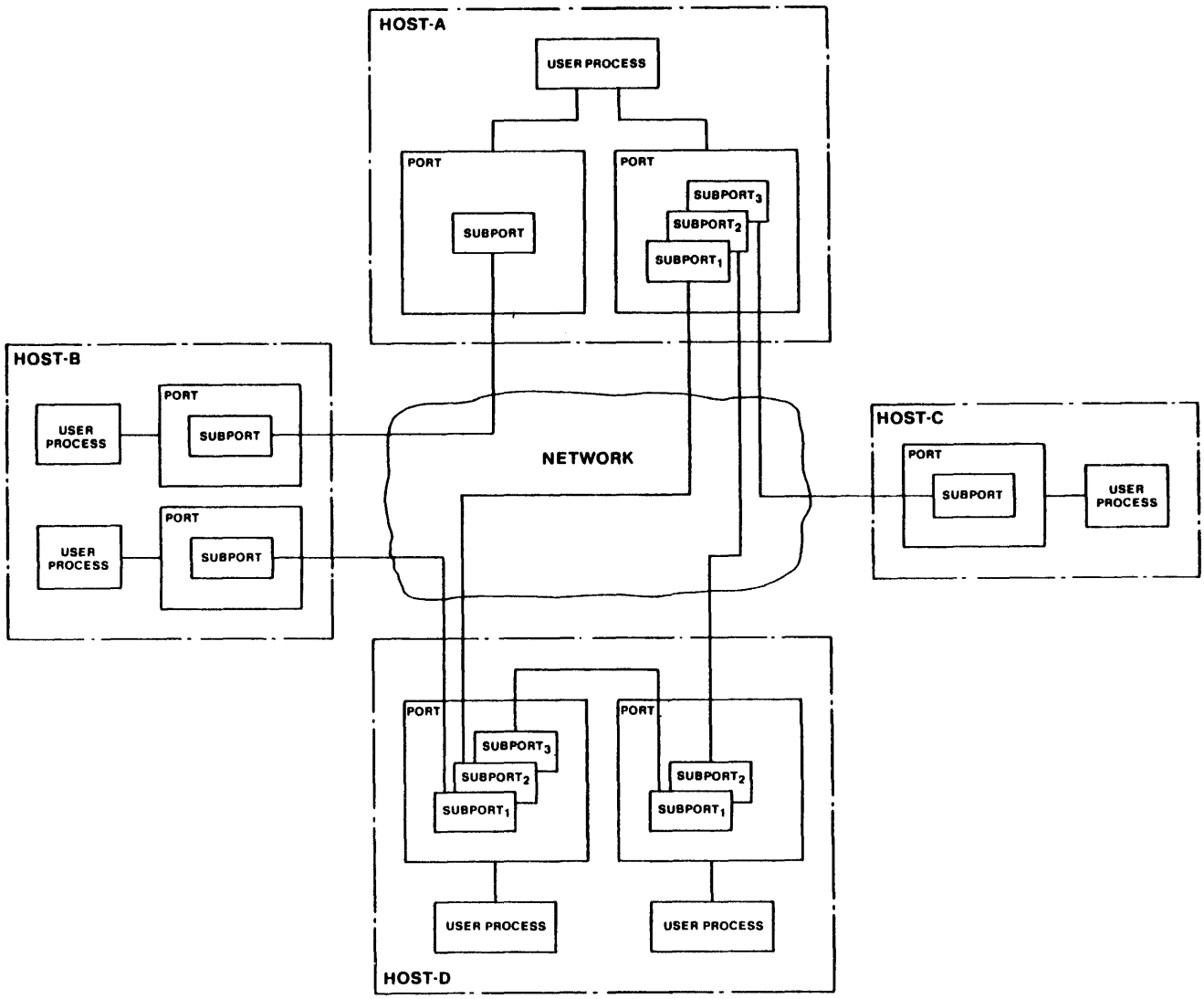


Figure 5-3. Port and Subport Interconnection

Included in the Port are Subports (subfiles) each of which provides a unique connection to one end-point user. In fact, a Port can be regarded as consisting of one or more subports, where the Port provides the common user interface to a defined group of subports. Figure 5-3 shows the relationships between port and subports. Notice that end-points may be at the same host (local subports).

Each subport contains elements which are unique to its remote connection, such as the attributes describing the remote hostname, node address, port address, subport address, and usercode. It also contains local attributes such as its own subport address.

The Port contains elements that are common to all subports, such as Port Name. These are called Port attributes. Port attributes that can be changed during the existance of a Port will only affect subports opened after the change, existing subports will be unaffected.

Figure 5-4 shows a simplified representation of a port with subports and is referred to in the following discussions.

Many new terms that are introduced in the following discussions are in the form of attribute names, and functional states. The attributes are denoted as capitalized names, and their description can be found under Port Level Attributes later in this section. Functional states of the subports can also be found in the attribute section, and Figure 5-10 should be referred to for the state relationships.

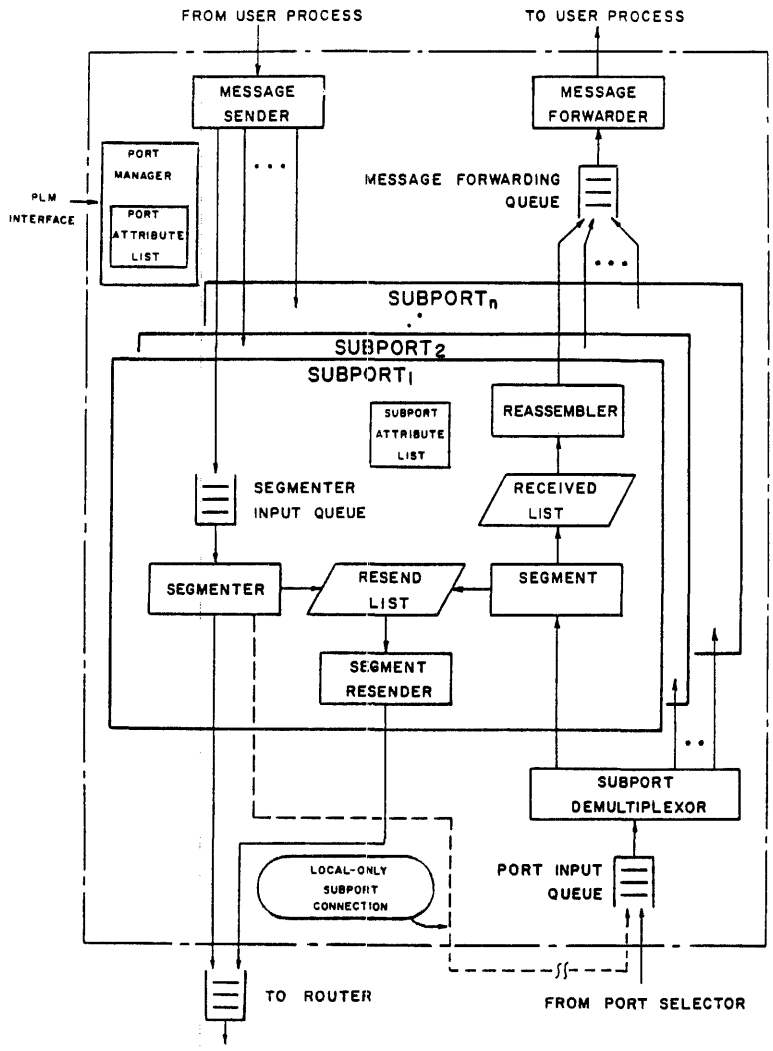


Figure 5-4. Port and Subport Organization

Data Transmission - Sending Messages

The user program WRITE function becomes the Send Message function in the port and subport. This section describes the techniques used to send user messages.

A user can only WRITE messages to the port and subport when the subport is OPENED; that is after the end-point subport has been found and is communicating.

NOTE

When a user program opens a port, the port and subport can proceed through several internal states before becoming OPENED, refer to Figure 5-10.

When a message written by a user process is obtained by the port and subport, it is forwarded to the end-point subport. The end-point may be either across the network or local to the host. (From a user viewpoint, a subport that is communicating with a subport in the same host appears the same as subports that are communicating in different hosts.)

The message to be sent by the port may be compressed, if desired, and segmented if required to obtain segments of a suitable size for network transmission. A PORT FRAME is built for each segment sent, and when flow control and reachability permit, it is forwarded to the Router with the destination node address. A copy of each PORT FRAME is kept in a segment resend list in case retransmission is required. Retransmission of lost, or bad segments is described further in the subsection, Segment Retransmission. Where possible, acknowledgments of received frames are sent with outgoing message segments.

The transmission of messages between ports may be affected by changes in network conditions such as:

BLOCKED:
This occurs when the path between subports is temporarily broken. All subports communicating with a particular host become BLOCKED at the same time. When this condition occurs, the PL BLOCKED TIMEOUT timer is started, and if it should expire, all subports connected to that host become DEACTIVATED.

Messages segments will be sent as soon as the condition BLOCKED is removed.

SUBPORT CLOSURE:
Closure of the subport only occurs due to an explicit CLOSE by the user or system.

System closure can occur if the user program is "discontinued" due to an explicit request or due to a user program failure.

When the user CLOSES the subport, a CLOSE REQUEST Frame is generated by the subport and sent to the end-point subport. When the frame is acknowledged, the subport state becomes CLOSED or NEVER OPENED depending on the terminate option associated with the CLOSE operation.

DEACTIVATION:
DEACTIVATION occurs due to a function that denies access to the remote end-point subport. This can occur due to the closure of the remote subport, or if an abort frame is received, or any cause that makes it appear to be unreachable, such as excessive errors, or unreachable host.

PENDING DEACTIVATION:
PENDING DEACTIVATION preceeds the state DEACTIVATION. The pending state remains as long as there are any input messages queued for the subport. Output messages are automatically flushed out of all port and subport buffers; they are not sent.

When there are no messages to send, the subport will wait. The subport remains in a condition to transmit messages until it is CLOSED or DEACTIVATED.

If the port condition of BLOCKED occurs, message segments cannot be sent from the subport to the end-point subport. However, the user program can continue to WRITE messages to the port and subport while there is room on the Segmenter Input Queue.

Any attempt by the user to send a message during a PENDING DEACTIVATION or DEACTIVATION state, results in an end-of-file action.

Data Transmission - Receiving Messages

After the matching process has been completed and an end-point subport is communicating, messages can be received by the subport. However, the user can only access input messages when the subport OPEN is completed, that is when the subport state is OPENED. One subport may have messages in transit however before the other subport has received and acted upon all Port Level Manager (PLM) messages. This feature allows messages to be sent and honored while a subport is finalizing its housekeeping before becoming OPENED.

NOTE

When a user program opens a port, the port and subport can proceed through several internal states before becoming OPENED, refer to Figure 5-10.

Messages, segmented into frames, traverse the network to arrive at the destination node. At the destination node, they are passed from the Station Level to the Router and then to the Port Selector. Each level checks its header, and removes it before passing the frame on. The frames arrive at the Port Level with only Port Level Headers.

In the Port Level, the Port Selector directs the frames to the desired port. In the port, the subport will resequence segments if delivered out-of-order, expand them if required, and reassemble them into a complete message which will be placed into the Message Forwarding Queue. The user is then informed that a message is waiting by the change of state of the INPUTEVENT attribute and the change of the number of input messages waiting, indicated by the MESSAGE QUEUE SIZE attribute. The complete message is prepared for the user, and all headers are removed by the subport before being placed in the Message Forwarding Queue. To obtain the messages from the Message Forwarding Queue, a user program performs a READ operation.

Messages may be received at the rate at which the subport can handle the incoming frames. When the subport can no longer handle incoming frames, flow control procedures take effect to reduce the rate of receiving incoming frames. Flow Control is described later in this section.

Conditions which may affect messages being received are as follows:

SUBPORT CLOSURE:
Closing the subport causes the subport to send out a CLOSE REQUEST frame to the end-point subport. If there are any information frames to be sent from this subport then the CLOSE REQUEST frame is sent out after the last information frame. When this CLOSE REQUEST frame is acknowledged, all the subport buffers are flushed, and the subport is CLOSED. During the period after the user CLOSE was issued, no further messages can be sent or received.

DEACTIVATION:
Receipt of a CLOSE REQUEST from the end-point subport causes the subport to enter a SUBPORT STATE of DEACTIVATION or PENDING DEACTIVATION, depending on whether there are any messages in the Message Forwarding Queue for the user.

The subport remains in a DEACTIVATION state until it is CLOSED. While in a PENDING DEACTIVATION state, any messages remaining in the Message Forwarding Queue can be obtained by the user.

Message Segmentation

Segmentation is used to limit message sizes to a manageable size for network transmission. In Network Services the subports are the only elements that segment messages.

The segment size used by the subport for a particular destination node is determined from the Remote-Hosts table kept by the PLM. This value is passed to the PLM by the Router, via the Network Services Manager (NSM), at the time that the Router gets knowledge of a remote host.

The values for segment size are established by the Router. Refer to the Router subsection for further information on segment size determination.

USE OF SEQUENCE NUMBERS

To allow multiple port frames to be in transit between origin and destination ports simultaneously, each port frame is assigned a unique sequence number. This number (in the frame defined as Ns) is used to re-order frames upon receipt and to detect missing or duplicated frames.

Within a subport, several functions and attributes interact to provide reliable port frame delivery. The attributes are: LAST NR SENT, LAST NS RECEIVED, LAST NS SENT, LAST NR RECEIVED, FLOW STATUS SENT, FLOW STATUS RECEIVED, WINDOW SIZE, SEGMENT TIME OUT, and RETRY LIMIT.

RETRY LIMIT, WINDOW SIZE and SEGMENT TIME OUT are derived from PLM attributes PL RETRY LIMIT, PL WINDOW SIZE and PL SEGMENT TIMEOUT - the current values are passed to the subport at subport OPEN time. The PLM attributes can be changed by the Operations Interface, but any existing subports are unaffected. The others may only be seen within the frames themselves and cannot be altered by the user.

FLOW CONTROL

Flow Control allows the receiving subport to control the transmission of data by the sending subport.

Flow Control between subports is handled by the window mechanism and by two flow status attributes (not accessible to the user). The window mechanism allows only a pre-determined number of INFORMATION PORT FRAMES to be in transit at any given time. The sending subport may not send any INFORMATION PORT FRAME whose sequence number is not between the LAST NR RECEIVED and the sum of LAST NR RECEIVED plus WINDOW SIZE.

The flow status attributes are used to slow the sender when the receiver's using process is not handling messages fast enough. These attributes are communicated between the subports by the RECEIVE READY and RECEIVE NOT READY SUBPORT CONTROL FRAMES. These frames are also given sequence numbers to order them and guarantee their arrival, but these numbers are independent of the information sequence numbers.

The flow status attributes are set to indicate a "not ready" condition whenever the Message Forwarding Queue exceeds its limit size. It is set to indicate a "ready" condition whenever the number of entries in the Message Forwarding Queue falls below a threshold identified by the RESUME READY FACTOR. The RESUME READY FACTOR is a percentage applied to the message queue limit to determine the threshold value. RESUME READY FACTOR is derived from the PLM attribute PORT RESUME READY FACTOR and the current value is passed to the subport at subport OPEN time. PORT RESUME READY FACTOR can be set via the Operations Interface and only applies where communication is between ports in different nodes. Local-only subports also have a RESUME READY FACTOR but it is not operator settable.

Sending Acknowledgments

The local subport returns acknowledgments for received segments to the remote subport under these conditions:

1. Whenever segments are sent to the remote subport:
   the LAST Ns RECEIVED is included in each segment as the new Nr value. The Nr frame field contains the 8 least significant bits of the last Ns received (0-255).

2. Whenever the local subport detects a DEMAND ACKNOWLEDGMENT INDICATOR in a received PORT FRAME:
   if there is no segment ready to be sent, an acknowledgment frame (SUBPORT CONTROL FRAME, type: UNNUMBERED ACK) is sent.

3. Whenever the receive window is half full:
   if there is no segment ready to be sent, an acknowledgment frame is made up and sent.

Segment Retransmission

Each PORT FRAME sent from a subport to the Router has a timeout value associated with it, SEGMENT TIMEOUT. When this time limit expires before acknowledgment is received, the PORT FRAME is assumed to be lost in the Network.

The first such "lost" PORT FRAME is resent to the Router after setting the DEMAND ACKNOWLEDGMENT INDICATOR. This retransmitted segment has the same SEND SEQUENCE NUMBER (Ns) as the original segment but the RECEIVE SEQUENCE NUMBER (Nr) reflects the current value of LAST NS RECEIVED. No other timed-out frames are resent until an acknowledgment is received.

After an acknowledgment is received, if there are PORT FRAMES that are still considered lost, the first such PORT FRAME is resent.

The attribute RETRY LIMIT determines the number of unsuccessful retransmissions of a particular segment before the subport is placed into a DEACTIVATED state, and the error, DISCONNECTED, is returned to the user program. Any messages that had not been sent from the subport prior to DEACTIVATION, will be lost.

Data Compression

The purpose of data compression is to achieve information transfer using the least number of data bytes. Data compression utilizes the links and transit nodes more efficiently, providing for greater information transfer across the network.

BNA provides compression at the Port level.

To use compression a subport must have the attribute COMPRESSION POSSIBLE true. COMPRESSION POSSIBLE is set true during PLM negotiations for subport connection, if both PLM's have the attribute PL COMPRESSION ALLOWED true. (PLM negotiation is described later in this section.)

When the user indicates that compression is to take place, by the subfile attribute COMPRESSION, a CHANGE COMPRESSION SUBPORT CONTROL frame is sent out to the remote subport indicating that compressed data will be sent in subsequent frames. Upon receipt of a CHANGE COMPRESSION CONTROL frame, the receiving subport sets its RECEIVING COMPRESSED DATA attribute and will expand subsequent information frames.

A subport cannot invoke the receipt of compressed frames. Only the sending user program can cause compressed data to be sent. To send compressed data in both directions requires actions by both communicating programs.

Compression, as implemented in BNA, replaces all occurences of four (4) to 255 contiguous identical characters or one or more contiguous "escape" (EBCDIC "27") characters by the following sequence: the "escape" character, a one byte (binary) character count, and the compressed character, e.g.:

"E8""C1""C1""C1""C1""C1""C2" replaced by "E8""27""05""C1""C2"

(Y A A A A A B) <---------- (Y "escape" 5 A B)

The compressed data sequence will not be split across a segment boundary.

PORT LEVEL MANAGER (PLM)

The Port Level Manager performs five types of overall level management.

First, the Port Level Manager participates, in response to Commands from the Network Services Manager, in the process of initializing BNA Network Services at the local Host/Node. This includes obtaining the relevant initial settings of various attributes, sizing arrays, and general setup of the Port Level Manager's software environment.

Second, the Port Level Manager is involved, again in collaboration with the Network Services Manager, in the establishment (and validation) of communications with a remote Port Level Manager.

Third, the Port Level Manager establishes and terminates its own subport dialogs with remote port level managers. It has its own port for these communications, and that port contains one subport for each remote PLM.

Fourth, the Port Level Manager interfaces with the Network Services Manager to handle the specification, modification, and control of the BNA Network configuration as seen at the Port (Host) Level.

Fifth, the Port Level Manager participates, under the control of the Network Services Manager, in the process of shutting down the local Node, i.e., disconnecting it from the Network.

Port Level Tables

The Port Level Manager maintains three major tables: The Remote Hosts Table, the Candidates-for-Match List, and the Allocated Ports List. These are described in the following paragraphs.

THE REMOTE HOSTS TABLE

This list contains an entry for each remote host in the network of which the local PLM is aware. Each remote host is identified by its host name and the associated node address.

The information for this table is communicated to the PLM by the Router, the Operations Interface, and the remote PLMs.

THE CANDIDATES FOR MATCH LIST

This list contains an entry for each subport for which the PLM is attempting to establish a subport dialog. Each entry may be for a local or remote subport. Each entry specifies attributes and addresses of the subport that may be needed by the matching algorithm to match subports.

The information contained in this list is provided by local OPEN commands and by requests from remote PLMs for connections to subports at the local host.

THE ALLOCATED PORTS LIST

This list enables the PLM to determine whether a subport being opened belongs to an already allocated port, or is one that must be newly-established. It contains information to make this determination.

Subport Dialog Management

This section discusses the functions of the Port Level that are used to establish end-point subport communication. This includes the process of finding end-point subports, called Matching, and the actual creation of the end-point subports.

MATCHING

Matching is the function of finding the desired subports for communication between two user programs. Matching occurs when a subport OPEN initiates PLM functions to find a complementary end-point subport. The method used by the PLM to match subports varies according to the subport OPEN option. There are three options for OPEN: WAIT, RETURN, and AVAILABLE.

The WAIT and RETURN options cause the PLM to attempt to find a match, and the OPEN will remain waiting until a match is found, or the subport is CLOSED. (Note: RETURN corresponds to the user interface option, OFFER).

The AVAILABLE option will only result in a match if the PLM is already aware of a complementary end-point subport. If there is no end-point subport, a NO FILE FOUND result is returned. Another explicit subport OPEN must be performed if it is desired to open this subport later.

Matching Responsibility

The process of matching demands that one of the PLMs has the final decision to allow or not allow matching. This is termed Matching Responsibility, and it is granted to the PLM of the node with the smaller node address.

Only the PLM with Matching Responsibility may send the MATCH message.

Candidate

A candidate is a potential subport. It is created when a user program requests the subport opened. It remains a candidate until it is connected to a remote subport (the ALMOST OPENED state), or until it is CLOSED.

Subport Matching

The Candidates-For-Match-List is a chronologically ordered linked list with entries for local and remote subports seeking a match. A candidate is added to the list whenever:

- The PLM receives a subport OFFER message from a remote PLM or
- The PLM receives a subport OPEN request from a user. If the OPEN requests that more than one subport of a port be opened, multiple entries may be added to the list.

A candidate is added to the end of the list and is compared with all candidates already on the list. Candidates A and B match if, and only if, all of the following conditions are true:

- PORTNAME A = PORTNAME B
- YOURNAME A = NULL OR YOURNAME A = MYNAME B
- YOURNAME B = NULL OR YOURNAME B = MYNAME A
- YOURHOST A = NULL OR YOURHOST A = MYHOST B
- YOURHOST B = NULL OR YOURHOST B = MYHOST A
- Suitable security qualifications are met.
- Both candidates are not subports of the same port.

NOTE

For matching purposes of the attributes PORTNAME, YOURHOST, and MYHOST, upper or lower case letters, and dash or underline are considered equivalent. E.g., (A = a) and (- = _).

Subport Matching Algorithm

- If a matching candidate is found and both are local, the candidates are removed, and the OPEN is completed.

- If a matching candidate is found and one of the candidates is remote, and the local PLM has Matching Responsibility, a MATCH message is sent to the remote PLM. If the remote PLM approves of the match after a security check, it sends back an ACCEPT MATCH message. When an ACCEPT MATCH is received by the PLM, the candidates are removed, and the OPEN is completed.

- If a matching candidate is found and one of the candidates is remote, and the local PLM does NOT have Matching Responsibility, and an OFFER has not been sent, a JUDGE QUICKLY OFFER is sent to the remote PLM. If the remote PLM finds a match for the OFFER, it returns a MATCH message. When the MATCH is received by the local PLM it sends an ACCEPT MATCH (security first being checked), the candidates are removed, and the OPEN is completed.

- If no matching candidate is found for a local candidate with OPEN type of AVAILABLE and a NULL remote hostname, the candidate is removed and the user program is returned the reason for no match, NOFILEFOUND.

- If no matching candidate is found for a local candidate with OPEN type of AVAILABLE and a specified remote hostname, a JUDGE QUICKLY offer is sent to the remote PLM.

- If no matching candidate is found for a local candidate with OPEN type of WAIT or OFFER (RETURN) and a NULL remote hostname, it passively waits on the list for another candidate to match it.

- If no matching candidate is found for a local candidate with OPEN type of WAIT or RETURN and a specified remote hostname, an OFFER, type BE PATIENT, is sent to the remote PLM.

- If no matching candidate is found for a remote candidate with offer type of JUDGE QUICKLY, a NO MATCH message is sent to the remote PLM.

- If no matching candidate is found for a remote candidate with offer type of BE PATIENT, it passively waits on the list for another candidate to match it.

- If a MATCH is received that cannot be honored, a REFUSE MATCH message is sent. This occurs if the security qualifications are not met or the offer is no longer valid (due to subport closure, etc.).

YOUR NAME - MY NAME Relationship

To provide selectivity of ports and subports by users of subports, the attributes YOUR NAME and MY NAME are used. This allows the user to specify the precise end-point port for communication, by specifying the subport attribute YOUR NAME to be the same as the end-point port attribute MY NAME. Figure 5-5 shows the Port/Subport name relationships.

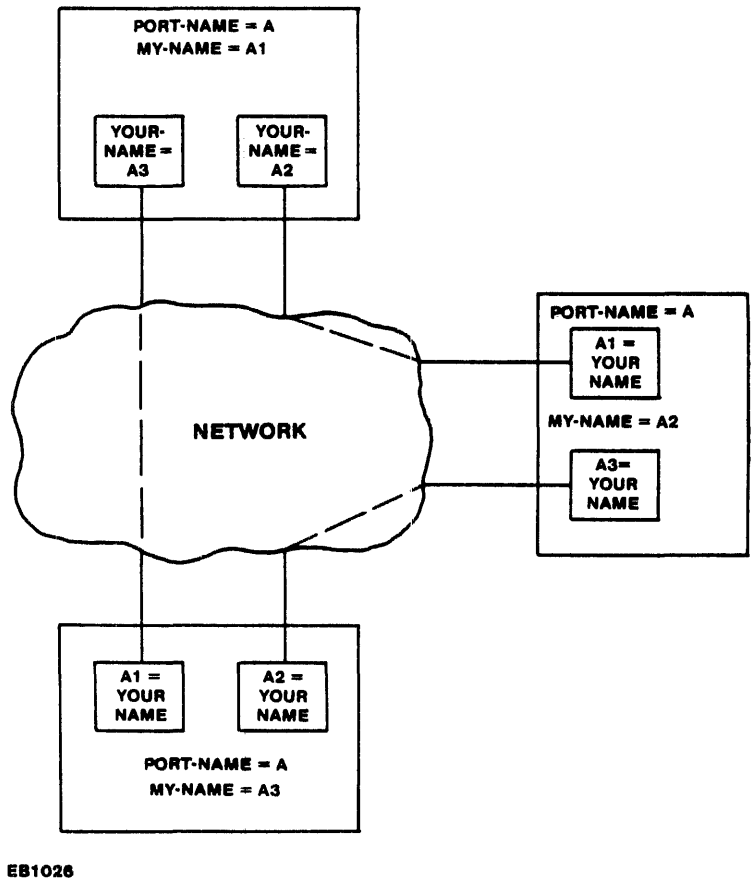
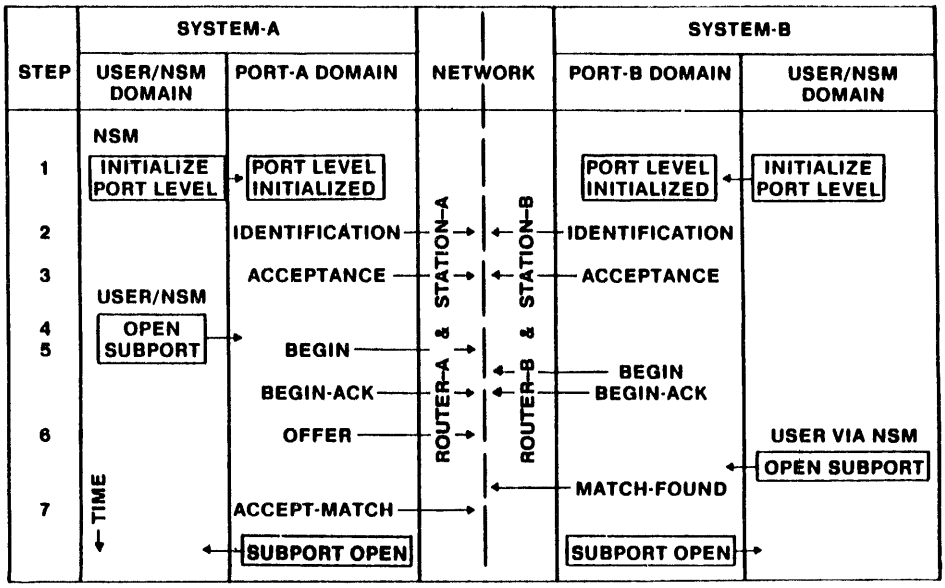


Figure 5-5. Port/Subport Name Relationships

SUBPORT CREATION

Figure 5-6 shows the steps and order of messages that flow between port levels of two remote nodes, from the creation of a port level at initialization time, to the completion of the connection of two subports. The following discussion will describe the steps in this process.



EB1027

Figure 5-6. Port Level Communications Between Hosts

To simplify the descriptions, the following notations will be used to distinguish PLM's of either node: PLM of node A will be called PLM A and PLM of node B will be called PLM B. Similarly, other terms will be appended with either A or B. If required, the frame section should be consulted for further detail on any mentioned frame.

Step 1. - Refer to Figures 5-6 and 5-7.
After initialization the PLM's have the capability of communicating across the network.

At some stage, either during or after initialization, the Router will become aware of another node. This node is discovered during Station and Router greetings.

The Router then informs the Port Level, via the NSM, of the remote node. For this discussion let Node-A commence first. Hence, Router A informs PLM A of the existance of Node B.

Step 2.

PLM A sends an IDENTIFICATION Frame to the Router for onward transmission to the remote node. Note that there is no PLM port yet; the PLM sends the frame directly to the Router.

The frame traverses the network to arrive at the destination node, Node-B. Router-B "processes" the frame and passes it up to Port Level-B and to PLM B.

PLM B checks the validity of the identification of PLM A and if valid sends out an IDENTIFICATION frame of its own (if it has not already done so) and sends out an ACCEPTANCE frame. This informs PLM A that it has been accepted for further dialogs.

Step 3.

PLM A receives the IDENTIFICATION frame from PLM B, checks its validity, and if valid sends out an ACCEPTANCE frame. (If it were not valid it would not respond).

At this stage both PLM's are aware of each other, but do not do anything else until a user requests the opening of a port and subport. Figure 5-7 shows the state at this time.

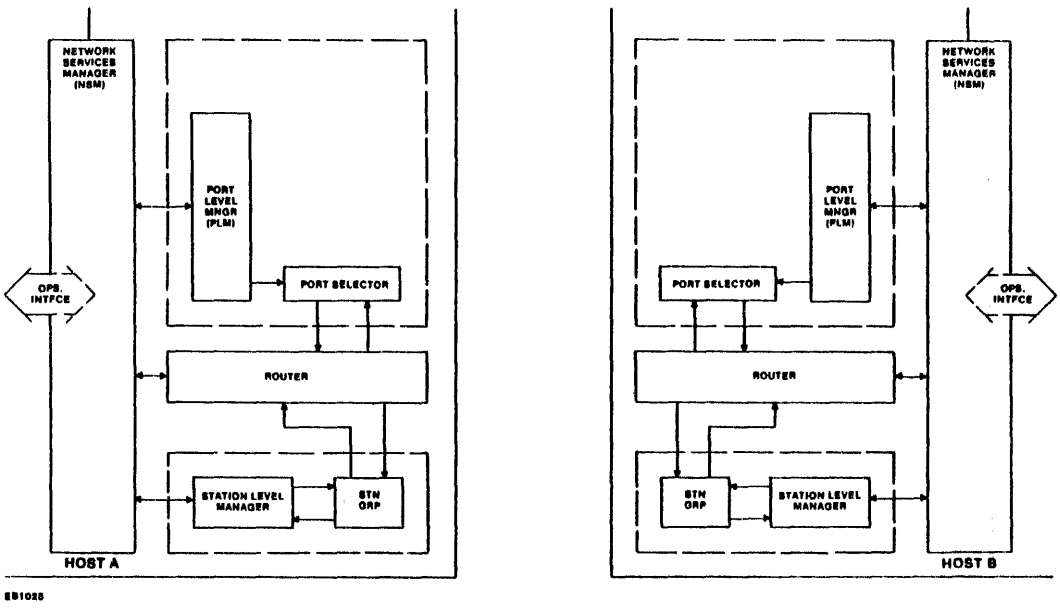


Figure 5-7. Identification and Acceptance

Step 4.

User-A sends an OPEN subport command to the PLM A. PLM A notices that it has no dialog capability to PLM B so proceeds to Step 5.

Step 5.

The PLM starts the creation of a port and subport for its own needs. Having assigned a PLM port and subport, the PLM sends out a BEGIN frame. This frame identifies the PLM's port and subport to PLM B.

PLM B receives the BEGIN frame from A, then like A, prepares a PLM port and subport and sends a BEGIN frame to PLM A. It also prepares and sends a BEGIN ACK frame to PLM A. The PORT MAX MESSAGE TEXT SIZE and PORT WINDOW SIZE are each set to the minimum of the corresponding field in the BEGIN A frame and the corresponding attribute of PLM B.

Similarly, PLM A, after receiving the BEGIN frame from B, will send a BEGIN ACK to PLM B.

At this time both PLM's have ports that can be used for messages.

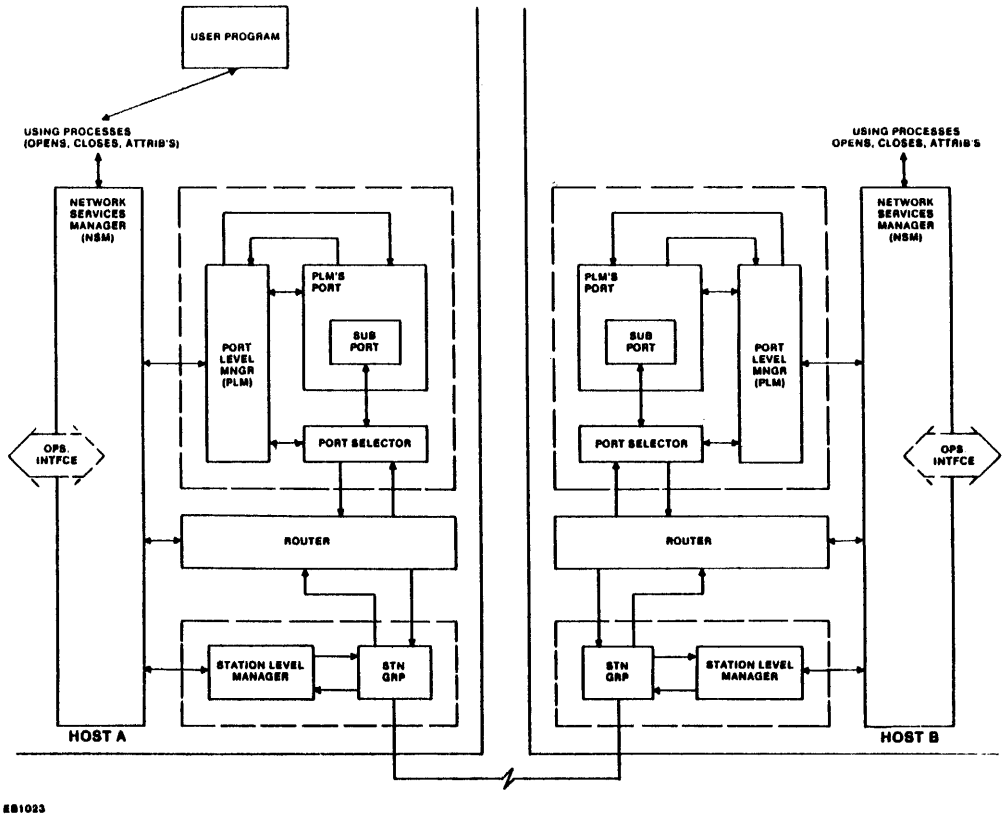


Figure 5-8. Establishment of Ports and Subports

Figure 5-8 shows the PLMs with their own port and subports. These subports are necessary to continue any further PLM communications as they perform segmentation of the longer PLM messages.

Step 6.

Having established PLM subports, the process of establishment of the subports for the user-requested OPEN begins.

PLM A checks the Candidates-For-Match-List and finds no Match. The details of the potential subport are entered into the Candidates-For-Match-List.

Matching takes place as described previously under Matching.

Step 7.

Finally, the ACCEPT MATCH is sent by PLM A and the subport is opened for use by the user.

If for any reason, such as the remote subport being non-existent, and the OPEN was of type WAIT or OFFER, no match would be found (at this time) and the subport requests would remain as candidates on each PLM's Candidates-For-Match list. These candidates will remain on the list until such time as a match is forthcoming or the user CLOSES the subport.

Figure 5-9 shows the final port level representation after the subports have been opened.

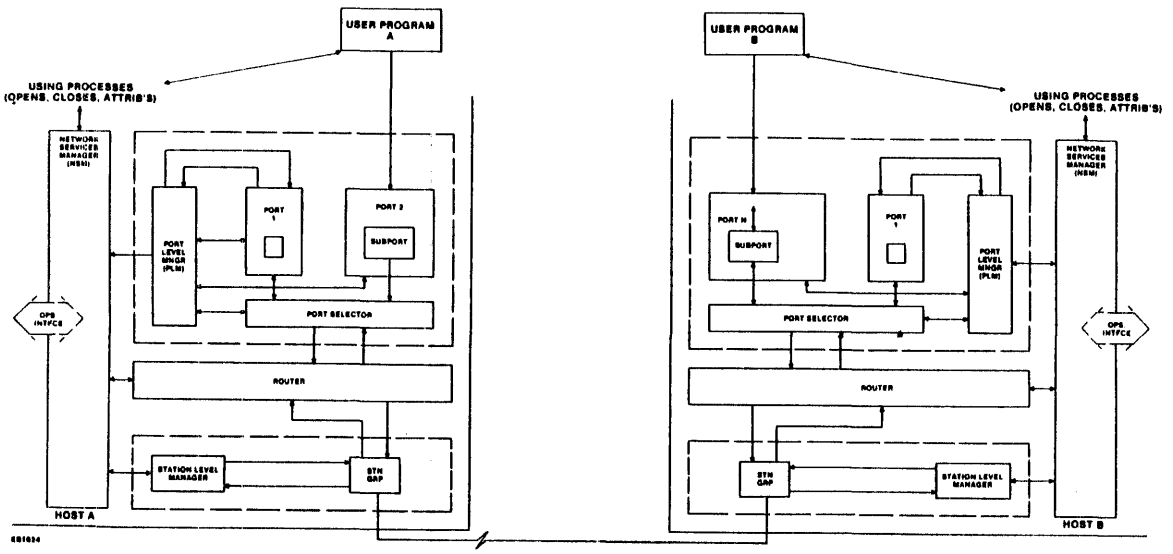


Figure 5-9. Opening of Subports

PORT LEVEL INTERFACES

Refer to Figure 5-2 for the Port Level interfaces.

The type of information that crosses these interfaces take the form of commands, responses to commands, and unsolicited reports.

Several commands must pass between a Port and Subport and any of the users, the PLM, and the Router. They support mechanisms for initiating and terminating conversations, for sending and receiving messages, for recovery from errors, and for monitoring operations.

There are several interfaces through which this information will pass: between the PLM and Port and Subport, between users (program or Host Services dialogs) and a Port and Subport, and between Port and Subport and the Router (via the NSM).

Port Level Manager with Port Interface

Commands that pass through this interface are used for setting up the ports and subports, and monitoring operations. These commands may originate: in the Router, the user program (both via the NSM); from the NSM itself; and from the PLM.

Port Level Manager with Subport Interface

Commands that pass through this interface are used for initiating and terminating Subports and Subport dialogs, for recovery from errors, and for monitoring operations.

User with Port Interface

The user may send commands to the Port to send and receive messages through a Subport. In these commands various options, such as WAIT, IMMEDIATE, etc., are used to indicate user response options. These are discussed further in Section 3.

NSM with PLM Interface

Across this interface pass commands and reports that originate at the user program, at the NSM, at the Router, Operations Interface, and the Port Level Manager.

User program commands include changes to port and subport attributes, opening subports, and closing subports.

Router originated commands inform the Port Level Manager of changes to Host and Network parameters, such as Host/node address information and segment sizes in use with remote hosts.

Operations Interface originated commands carry instructions entered by Network Services Agents or the initialization file.

Port Level originating traffic across this interface includes: reports, such as closing port information, used for logging port resource usage; reports of Port Level errors; and responses to commands.

Router with Port Level Interface

This interface is used for the transmission/reception of Port Level segments to/from the network via the lower levels of Network Services: Router and Station. All network traffic originating or terminating in the Port Level crosses this interface. This interface consists of two queues, one queue for each direction.

HOST SERVICES AND THE PORTS

Host Services uses Network Services as its transport mechanism for supporting the Host Services dialogs. It uses ports in the same manner as any other port user, with one exception - the port level recognizes OFFERS for Host Services port connection by the special usercode of hex "BADBAD", a usercode not available to other users. If a match cannot be found for the OFFER, the Port Level Manager communicates to the local Host Services process to open a suitable port for the dialog. Closing Host Services ports is accomplished in the normal manner.

PORT LEVEL ATTRIBUTES

The following paragraphs describe the attributes at the Port Level. These include Port Level Manager attributes, Port attributes, and Subport attributes.

Port Level Manager Attributes

AUTO PLM DIALOG TERMINATE

This attribute, if true, indicates that a PLM-to-PLM dialog is to be terminated automatically if, for a continuous time interval determined by the PLM DIALOG TIMEOUT attribute there is no subport allocated for use with the associated remote Host on behalf of any using process (including user programs and Host Services processes).

This attribute can be set via the Operations Interface command HOSTINACTIVEDISCONNECT.

COMMUNICATIONS STATE

This attribute indicates the current state of the PLM-to-PLM communications with a given remote host. The states are:

ACTIVE:

There is a PLM-to-PLM dialog established and open.

ESTABLISHING:

An attempt to establish a PLM-to-PLM dialog is underway. The local host has either sent or received a BEGIN control frame.

GREETING:
A PLM IDENTIFICATION control frame has either been sent or received, but the exchange of greetings has not been completed.

INTERRUPTED:
The remote host has been validated, but has become unreachable; this will require validation to be repeated when the remote host again becomes reachable.

QUIET:
The remote host has been validated, and the exchange of greetings has been completed, but no PLM-to-PLM dialog is in progress, nor is there an attempt to establish one underway.

TERMINATING:
A TERMINATE PLM DIALOG message has been sent to the remote host, but the associated PLM's subport is still open.

UNAVAILABLE:
The local Port Level Manager considers the remote host to be unavailable for communications.

HOST NAME
This attribute is the name of the local Host. It can be set via the Operations Interface HN command.

HOST VALIDATE
This attribute is determined at Node Initialization time. If it is True, then a [Host Name, Node Address] pair reported as reachable by the Network Services Manager or received in a PLM IDENTIFICATION control frame is accepted as valid if and only if it has already been specified as valid in an ADD HOST command. If it is False, such a pair is valid if and only if neither element of the pair appears in any other pair in the REMOTE HOSTS table.

This attribute is set by the Operations Interface command VALIDATE.

INCARNATION ID
This attribute identifies the incarnation of the PORT LEVEL MANAGER that is currently running on the local Host.

This attribute cannot be accessed by a user.

NODE ADDRESS
This attribute is the address of the local Node in the Network. It can be set via the Operations Interface LOCALIDENTITY command.

PL CONTROL RETRY LIMIT

This attribute specifies the maximum number of times the PORT LEVEL MANAGER attempts to resend a Control Frame that has timed out, that is, for which no response or acknowledgment has been received within the time specified by the PL CONTROL TIMEOUT attribute.

This attribute derives its value from PL RETRY LIMIT.

PL CONTROL TIMEOUT

This attribute is the maximum time the Port Level Manager waits for a response or acknowledgment to a control frame that has been sent out.

This attribute derives its value from PL SEGMENT TIMEOUT.

PL LOGGING INTERVAL

Refer to the Operations Interface command PORTLOGINTERVAL.

PLM-PHASE

This attribute is used to coordinate the initialization of the Port Level with the rest of Network Services. Values for this attribute are: NASCENT, ISOLATED, ATTRIBUTE ENTRY, CONFIGURATION ENTRY, OPERATING, SLOW SHUTDOWN, FAST SHUTDOWN.

Nascent phase occurs before the Port Level has any functional capabilities.

Isolated is the mode where only local ports can communicate; that is before a NET+ command has been entered.

Attribute Entry and Configuration Entry are phases entered during initialization (processing of a NET+ command).

Operating is after initialization has completed and indicates the normal operation of the host connected to the network.

Slow Shutdown is the mode after a NET- command has been entered to shut down the node. Only after every subport has been closed, or is in a deactivated state and there are no more acknowledgments to be sent, does the node complete its shutdown and the phase reverts to Isolated.

Fast Shutdown occurs after a NET- NOW command has been entered. This causes the Port Level to immediately deactivate all subports and enter Isolated mode.

PLM BLOCKED TIMEOUT

This attribute specifies how much time the Port Level Manager waits after a remote host becomes unreachable before (forcibly) DEACTIVATING all subport dialogs in progress with that host. (While the remote host is unreachable, any subport involved in such a dialog has a SUBPORT STATE of BLOCKED.)

This attribute can be set via the Operations Interface command HOSTUNREACHABLETIMEOUT.

PLM DIALOG TIMEOUT

This attribute is the amount of time a PLM-to-PLM dialog is maintained without any subports allocated for use by using processes for communication with a given remote Host. It is only meaningful if the AUTO PLM DIALOG TERMINATE attribute is true.

This can be set via the Operations Interface command HOSTINACTIVETIMEOUT.

VERSION

VERSION ID, VERSION PROTOCOL, VERSION COMPATIBILITY are attributes that describe the particular characteristics of the port level software being used. These attributes are used by the PLM for its identification procedures with other PLMs. They cannot be set or changed, other than by installing another software version.

PORT AND SUBPORT ATTRIBUTES

The Port Level Manager maintains several attributes that are used to set the values of the corresponding port and subport attributes during a subport OPEN sequence. Thus setting these attributes affects only subports opened after the change. These attributes are:

PL COMPRESSION ALLOWED

This attribute indicates whether the Node is willing to perform data compression/expansion for dialogs with remote hosts.

This attribute can be set using the Operations Interface command PORTCOMPRESSIONALLOWED.

PL RESUME READY FACTOR

There are two type of Resume Ready Factor: Local and Remote. The Remote type is used for subports in communication with other hosts. The Local type is used only for local subport dialogs (i.e., both subports are in the same host). Only the Remote type is accessible to Network Services.

The attribute can be set for subports with remote conversations via the Operations Interface command PORTRESUMEREADY.

PL RETRY LIMIT

This attribute is used to set the subport attribute RETRY LIMIT. The RETRY LIMIT attribute of a subport specifies the number of times to retransmit any particular segment before setting the SUBPORT ERROR attribute to DISCONNECTED and INFERRING DEACTIVATION.

This attribute can be set using the Operations Interface command PORTRETRYLIMIT.

PL SEGMENT TIMEOUT

This attribute is used to set the subport attribute SEGMENT TIMEOUT. The SEGMENT TIMEOUT attribute of a subport specifies the time to wait for acknowledgment of any particular segment before attempting to retransmit it or reporting an error.

This attribute can be set using the Operations Interface command PORTSEGMENTTIMEOUT.

PL WINDOW SIZE

This attribute is used to set the subport attribute WINDOW SIZE. The WINDOW SIZE attribute of a subport indicates the maximum number of segments that a subport may send (and the number it may expect to receive) that are unacknowledged.

This attribute can be set using the Operations Interface command PORTWINDOWSIZE.

Port Attributes

The port attributes are described here. These attributes are replicated for each port.

Note that those port attributes set by the user program retain values to which the user set them across subport CLOSES and subsequent re-OPENS. Attributes are not retained when a port or subport go to the state of NEVER OPENED, refer to Figure 5-10.

When attributes of the port are not described here, or additional information is required, other descriptions can be found in Section 3.

CHANGED EVENT

This event is caused whenever the STATE EVENT attribute is happened for any of the subports of this port. It is reset when this is no longer true. This attribute is the CHANGEEVENT file attribute.

INPUT EVENT (port)

This event is caused whenever a message is place into the Message Forwarding Queue, and it is reset when this queue becomes empty. It is the file attribute INPUTEVENT.

INTNAME

The INTNAME of a port variable is the name by which it is declared and referenced by the program which is using it. It is the file attribute INTNAME, of the file of type PORT.

This attribute is set when the user opens a subport of the port.

LAST SUBPORT USED

The LAST SUBPORT USED attribute indicates which subport was last used in a successful Send Message or Receive Message operation. If the last successful operation was a Receive Message with the subport index of zero, this attribute has the subport index of the subport that provided the message. Following a successful broadcast Send Message operation, this attribute has the value zero. Following an operation that was not successful, the value is unchanged.

This is the file attribute LASTSUBFILE.

MAX MESSAGE TEXT SIZE

This defines the limit on the size of a message within a port. It can be set by the user interface attribute MAXRECSIZE. Typically systems will have a system default MAX MESSAGE TEXT SIZE, and this may not be exceeded.

MAX SUBPORTS

The MAX SUBPORTS attribute defines the maximum number of subports a port may have open at any given time. It is the file attribute MAXSUBFILES.

This attribute can be set by the user only when no subports are open.

MESSAGE QUEUE SIZE (port)

The Message Queue Size of a port gives the number of messages queued in the Message Forwarding Queue for the port (they have not yet been received by the user process). It is the sum of the Message Queue Sizes for each subport in the port. This is the CENSUS file attribute.

MY HOST NAME

The MY HOST NAME attribute contains the name of the Host at which the process using the port is executing. It is used by the Port Level Manager in opening subport dialogs.

This attribute is set when the user opens a subport.

MY NAME

The MY NAME attribute is used to identify a port. It is used by the Port Level Manager in opening subport dialogs.

This attribute can be set when no subports of this port are open.

MY PORT ADDRESS

The value of the MY PORT ADDRESS attribute is used by Network Services to identify the local port.

PORT NAME

The PORT NAME of a port is the external name by which ports in communication with each other are associated. It is the file attribute TITLE.

The default value of PORT NAME is the same as the INTNAME of the port file. This attribute can be set when no subports of this port are open.

RESUME READY FACTOR

This attribute is used to produce a hysteresis effect for changing the flow status from not-ready to ready. This effect is obtained by using the RESUME READY FACTOR applied (as a percentage) against each resource limit to determine a threshold. The resource utilization controlled by the limit must fall below the threshold before the flow status may be set to READY.

This attribute is derived from the port attribute PL RESUME READY FACTOR, and is set when the first subport in a port is opened.

Subport Attributes

These attributes, which are replicated for each subport, allow for proper acknowledgments, FIFO reordering, and flow control of the segments.

Those subport attributes set by a user program retain the values to which the user set them across subport CLOSES and subsequent re-OPENS. Attributes are not retained when a port or subport go to the state of NEVER OPENED, refer to Figure 5-10.

When attributes of the subport are not described here, or additional information is required, other descriptions can be found in Section 3.

ACTUAL MAX MESSAGE TEXT SIZE

The ACTUAL MAX MESSAGE TEXT SIZE attribute specifies the value of MAX MESSAGE TEXT SIZE agreed to by the subports in this subport dialog. (MAX MESSAGE TEXT SIZE is a Port attribute). It is the MAXRECSIZE (sub)file attribute.

COMPRESSION POSSIBLE

The COMPRESSION POSSIBLE attribute indicates whether the local user may have the subport compress data or not. This attribute is related to the Port Level Manager attribute PL COMPRESSION ALLOWED. Refer to Compression discussion in this section.

FLOW STATUS RECEIVED

This attribute indicates whether the local subport has received a RECEIVE READY or RECEIVE NOT READY frame most recently. A RECEIVE NOT READY frame indicates that the local subport may not send any data segments (including re-transmissions) to the remote subport. It may, however, send its own RECEIVE READY, RECEIVE NOT READY, and acknowledgment frames. It also must send a CONTROL ACK frame acknowledging the receipt of the RECEIVE READY or RECEIVE NOT READY frame.

FLOW STATUS SENT

This attribute indicates whether the local subport has sent the corresponding remote subport a RECEIVE READY or RECEIVE NOT READY frame most recently. The local subport may accept incoming frames even if the FLOW STATUS SENT attribute is NOT-READY, because the remote subport may have already sent several (i.e., WINDOW SIZE) frames before receiving the RECEIVE NOT READY frame.

INPUT EVENT (subport)

The INPUT EVENT for a subport is caused whenever a message from this subport is placed into the port's MESSAGE FORWARDING QUEUE, and is reset when there are no more messages from this subport in this port's MESSAGE FORWARDING QUEUE. It is the (sub)file attribute INPUTEVENT.

LAST CONTROL NR RECEIVED

The LAST CONTROL NR RECEIVED attribute identifies the sequence number of the last control frame acknowledged to this subport.

LAST CONTROL NS RECEIVED

The LAST CONTROL NS RECEIVED attribute identifies the sequence number of the highest numbered (using modulo arithmetic) control frame received by this subport. Since the subport acknowledges control frames immediately, this is also the last control Nr sent.

LAST CONTROL NS SENT

The LAST CONTROL NS SENT attribute identifies the sequence number of the last CONTROL frame sent by this subport. (In this context, "last CONTROL frame" means the last frame that used a CONTROL SEQUENCE NUMBER.)

LAST NR RECEIVED

The LAST NR RECEIVED attribute of a subport identifies the sequence number of the last information segment acknowledged to this subport. The low-order 8 bits of the number (i.e. MOD 256) are contained in the frame; the entire number is maintained by the subport.

LAST NR SENT

The LAST NR SENT attribute of a subport identifies the sequence number of the last information segment acknowledged by this subport. The initial value of this attribute is zero (0).

LAST NS RECEIVED

The LAST NS RECEIVED attribute of a subport identifies the sequence number of the last good information segment received by this subport. In this context, "last good segment" means the last segment in sequence, with all previous segments in the sequence received, with no errors in this segment, and no uncorrected errors in previous segments.

LAST NS SENT

The LAST NS SENT attribute of a subport identifies the sequence number of the last segment sent by this subport. The initial value of this attribute is zero (0) but the attribute is incremented before being placed in the frame.

LOGGING INFO

This attribute collects usage statistics for logging and monitoring functions. Refer to Section 8 for further detail.

MAX SEGMENT SIZE

The MAX SEGMENT SIZE attribute indicates the maximum size of any segments of messages that may be sent via this subport. This MAX SEGMENT SIZE attribute does not include the sizes of the various (Router, Port, Subport) headers. It, therefore, differs from the MAX SEGMENT SIZE used by the Stations and from the one used by the Router.

The value for MAX SEGMENT SIZE is derived from the item in the entry for the remote host in the Port Level Manager's Remote Hosts Table PL MAX SEGMENT SIZE.

MESSAGE QUEUE SIZE (subport)

The Message Queue Size of a subport gives the number of messages from this subport currently queued in the Message Forwarding Queue for the port (they have not yet been READ by the user process). This is the (sub)file attribute CENSUS.

MY SUBPORT ADDRESS

The MY SUBPORT ADDRESS attribute is used by Network Services to identify the local subport within the local port.

OUTPUT EVENT

This attribute is caused whenever a Send Message with <send type> = IMMEDIATE operation would return with a value of OK. This means that there is room for at least one message on the Segmenter Input Queue. Send Message with <send type> = IMMEDIATE is the user function WRITE DONTWAIT.

RECEIVING COMPRESSED DATA

This attribute indicates whether the data will be received in a compressed form.

RETRY LIMIT

The RETRY LIMIT attribute of a subport specifies the number of times to retransmit any particular segment before setting the SUBPORT ERROR attribute to DISCONNECTED and inferring DEACTIVATION.

This attribute is derived from the Port Level Manager attribute PL RETRY LIMIT, and is set when the subport is opened.

SEGMENT TIMEOUT

The SEGMENT TIMEOUT attribute of a subport specifies the time to wait for acknowledgment of any particular segment before attempting to retransmit it or reporting an error.

This attribute is derived from the Port Level Manager attribute PL SEGMENT TIMEOUT, and it is set when the subport is opened.

SENDING COMPRESSED DATA

This attribute indicates whether the data will be sent in a compressed form. Changing the value of this attribute causes a CHANGE COMPRESSION control frame to be sent to the remote subport.

STATE EVENT

This attribute is called CHANGEEVENT as a (sub)file attribute. The STATE EVENT for a subport is caused whenever certain changes of the value of SUBPORT STATE occur. The STATE EVENT is reset whenever the user gets the value of the subport STATE attribute.

All changes of SUBPORT STATE cause the STATE EVENT except:

ALMOST OPENED <--> OPEN PENDING

NEVER OPENED <--> CLOSED

CLOSE PENDING <--> CLOSE BLOCKED

SUBPORT ERROR

SUBPORT ERROR indicates various error conditions of the subport. Refer to Section 3 for detail.

SUBPORT STATE

Figure 5-10 shows the subport state relationships. The values of the SUBPORT STATE attribute are listed after the figure. The values of subport state are most important to the user programmer, thus this information, combined with FILESTATE information of Section 3, should be consulted together for a complete understanding of this subject.

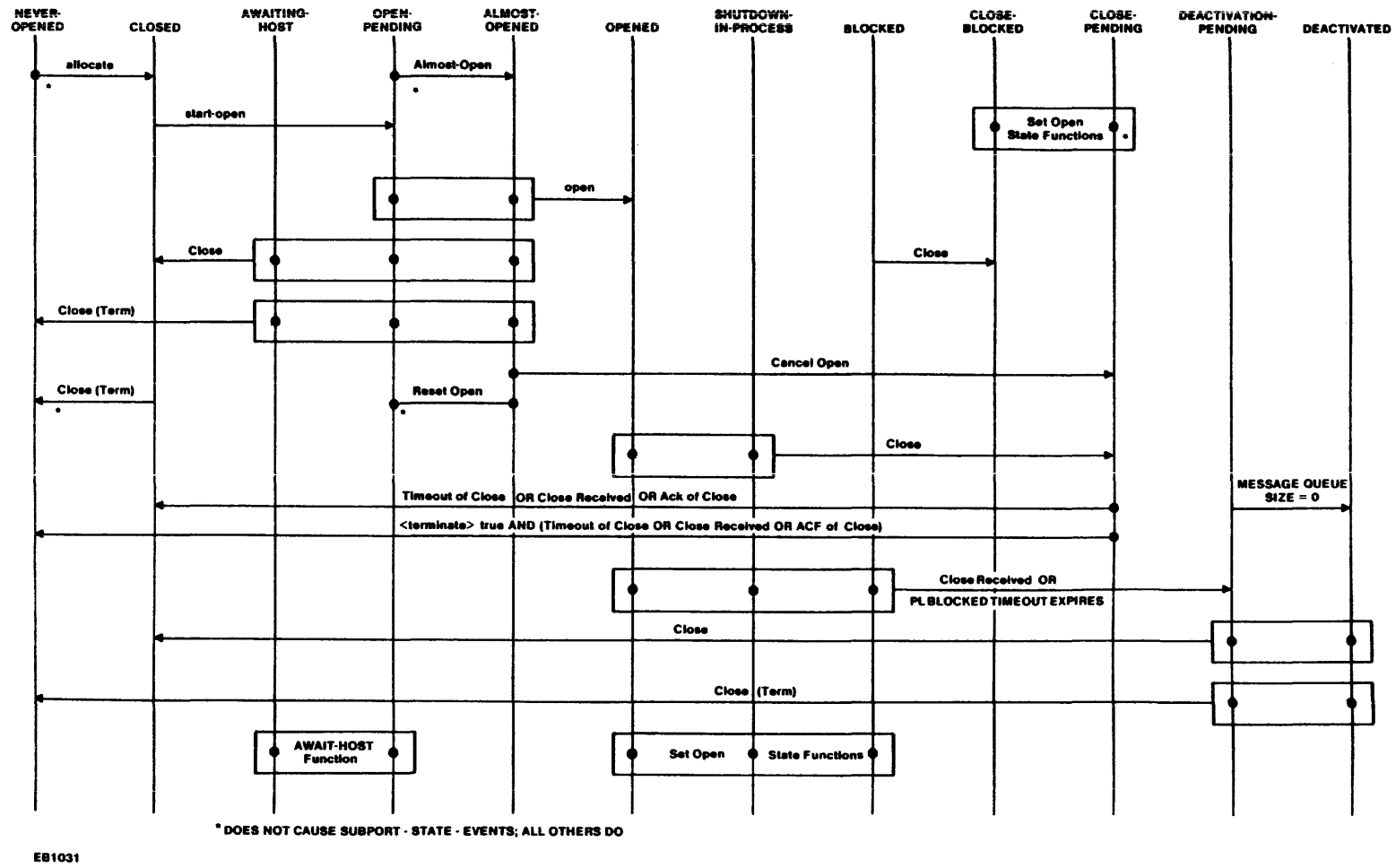


Figure 5-10. Subport State Transitions

ALMOST OPENED:
The OPEN operation is in progress. Segments may arrive from the remote subport.

AWAITING HOST:
A process has attempted to open the subport but the operation is not yet complete because the remote host is unavailable. The subport remains in this state until the host becomes available or the subport is CLOSED.

BLOCKED:
The remote host has become unavailable. The subport may be used for Send Message and Receive Message operations as long as resources are available.

CLOSE BLOCKED:
The subport is both CLOSE PENDING and BLOCKED. That is, the subport has been CLOSED by the local user and the remote host has become unavailable.

CLOSED:
The user program has completed a CLOSE operation on the subport. No messages are queued for the subport.

CLOSE PENDING:
The subport has been closed by the local user. All segments of messages previously sent by the user will be transmitted. When they have all been acknowledged by the remote subport, the SUBPORT STATE of the local subport becomes CLOSED or NEVER OPENED depending on <terminate>.

DEACTIVATED:
The subport either has received a CLOSE REQUEST or ABORT frame from the remote subport (indicating its desire to CLOSE) or has received an INFER DEACTIVATION command from its PORT LEVEL MANAGER or has attempted to resend some particular segment more than RETRY LIMIT times (indicating a disconnection) and the user of the local subport has received all messages from this subport. However, no retransmissions are done and any attempt to perform a Send Message or Receive Message operation using this subport results in an end-of-file action. When the local user closes the subport, the SUBPORT STATE of the local subport becomes CLOSED or NEVER OPENED depending on <terminate>.

DEACTIVATION PENDING:
This is similar to DEACTIVATION, with the exception that there are messages from the remote program which are queued for this subport (i.e., in the MESSAGE FORWARDING QUEUE). These messages remain available to be dequeued and processed. When no input messages remain in the queue, the state becomes DEACTIVATED. Any attempt to perform a Send Message operation using this subport results in an end-of-file action. If the local user closes the subport, the SUBPORT STATE of the local subport becomes CLOSED or NEVER OPENED depending on <terminate>.

NEVER OPENED:
Either the user program has not yet attempted to open the subport or the user program was CLOSED with <terminate> true. (In most languages, this is possible only by exiting the scope of declaration of the port.)

OPENED:
The subport may be used for Send Message or Receive Message operations.

OPEN PENDING:
The program has attempted to OPEN the subport but the operation is not yet complete.

SHUTDOWN IN PROCESS:
The local host is in the process of shutting down communications with this remote host. The subport may be used for Send Message or Receive Message operations.

TERMINATE
This attribute holds the value of the <terminate> parameter of a CLOSE command while the subport is waiting for an acknowledgment of its CLOSE REQUEST frame.

<terminate> parameter (Term)
When a user program closes a subport and does not exit the scope of the port declarations, a subsequent open on this subport will retain the old attributes. However, if the close does exit the scope of the port declarations, then the close has <terminate> true. A close of this nature results in the subport going to the state of NEVER OPENED. Any future open requests for this subport will create the subport with unspecified attributes being set to default values or conditions.

WINDOW SIZE
The WINDOW SIZE attribute indicates the number of segments that this subport may send (and the number it may expect to receive) that are unacknowledged.

The value for this attribute is negotiated during the Matching process. It is derived from the Port Level Manager attributes PL WINDOW SIZE of both end-point subports and is the smaller of the two values.

YOUR HOST NAME
The YOUR HOST NAME attribute specifies the host in the network which contains the remote program with which this local program wishes to communicate using this local subport. It is used by the Port Level Manager in opening subport dialogs. Its default value is the value of the MY HOST NAME attribute. This attribute may be set to NULL for subports which may be connected to any host. (In this case it will be set equal to the name of the actual remote host when the open is completed.)

YOUR NAME

The YOUR NAME of a subport corresponds to and is the same value as the MY NAME of the port with which it is communicating. It is used by the Port Level Manager in opening subport dialogs. Its default value is NULL.

YOUR NODE ADDRESS

The YOUR NODE ADDRESS attribute is used by Network Services to identify the destination Node for messages. The NODE ADDRESS must be unique for every Node in a given network. The mapping between HOST NAME and NODE ADDRESS is a function of Network Services.

YOUR PORT ADDRESS

The YOUR PORT ADDRESS attribute is used by Network Services to identify the remote port for messages.

YOUR SUBPORT ADDRESS

The YOUR SUBPORT ADDRESS attribute is used by Network Services to identify the remote subport for messages.

YOUR USERCODE

The YOUR USERCODE attribute of a subport indicates the usercode the user of the remote subport may use or is using. It is used by the Port Level Manager when setting up a subport dialog.

ROUTER

GENERAL

The primary functions of the Router are:

To route transit traffic from Station to Station,

To route locally originating traffic from Ports, and

To route locally terminating traffic to the Port Level.

Additional support functions include:

To respond to changes in the network and route accordingly.

To provide operational analysis tools, including Trace and Monitor.

A block diagram of the Router is shown in Figure 5-11.

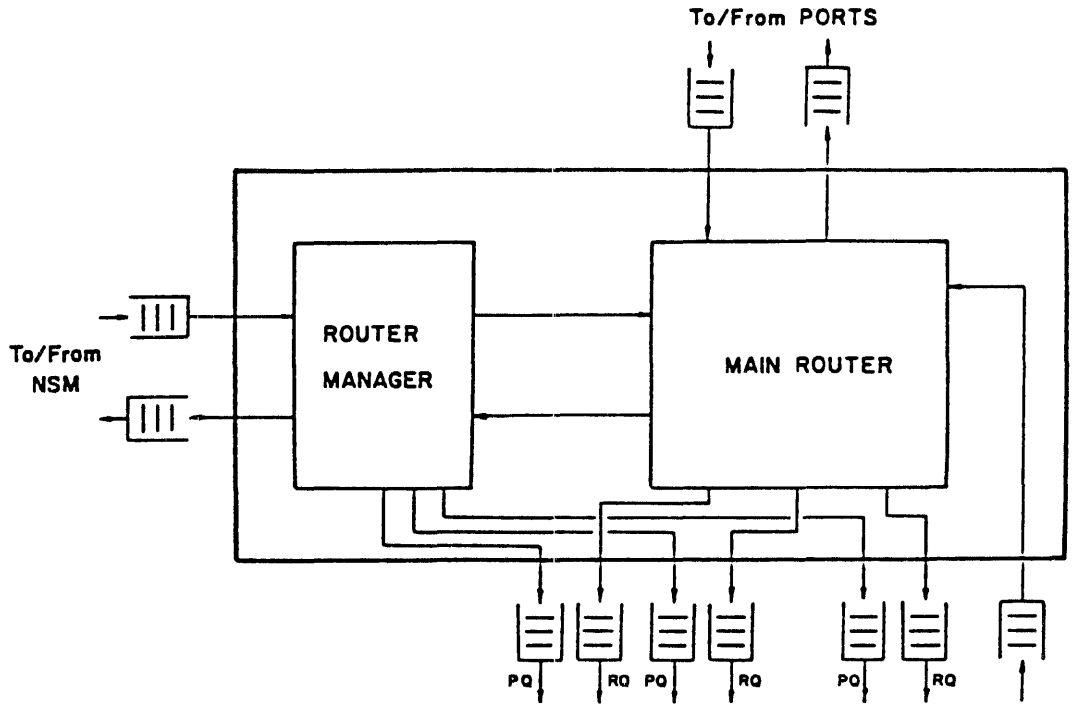


Figure 5-11. Router Level Block Diagram

ROUTING

The term "routing" is used in BNA in the usual sense of the selection of a path or channel for sending data. Routing is also used in a much more specific sense. A Routing defines the local node's view of one possible path for sending frames to a given Destination Node. It primarily identifies the neighbor node to which frames using this Routing are to be sent from the local node. This identification is in the form of the neighbor's Node Address and in terms of the queues from Router to Station Level. The Routing also includes some information about the path through the network to the Destination Node, including resistance and maximum segment size.

The BIAS routing update mechanism automatically takes into account the various capacities of heterogeneous nodes and links in a network. The capacity of each node is defined by a "node resistance". The capacity of each link is defined by a "link resistance". The BIAS mechanism finds "the paths of least resistance", which are the best paths by definition. Thus the system can "bias" the use of particular nodes and links in the network.

The mechanism automatically assigns default resistances to each node and to each link, based on capacity. Networks will work using these defaults throughout. In the case of multiple parallel links between two nodes, the routing mechanism uses a composite logical link resistance based on the combined capacity of the parallel physical links.

Provision is made, however, for the Operations Interface function to override the defaults in unusual circumstances. This manual setting of node and link resistances can be done at network initialization time, and while the network is operating. Mixtures of default resistances and resistances set by Operations are accommodated by the mechanism but may preclude future adaptive features. Manual override of the default resistances is not recommended.

Further control of the network is provided with a maximum resistance factor (MAXRF). The MAXRF is a Router attribute at each node and is normally the same throughout the network. Traffic is never routed over a path whose total resistance, the sum of the resistances of nodes and links along the path, is greater than or equal to MAXRF.

Network Size Limitations

The overall number of nodes in a BNA network is limited to 65534. The number of nodes which can be addressed by any frame is limited to 255.

ROUTER ATTRIBUTES

The following list identifies Router attributes.

LOCAL NODE ADDRESS
: The node address of the local node. Node addresses in a network must be assigned to nodes on a one-to-one basis. Node address 0 (zero) is invalid and 65535 (all ones) is reserved. Refer to the Operations Interface command LOCALIDENTITY.

TRANSIT COUNT LIMIT (TCNT LIMIT)
: A Transit Count (TCNT) is maintained in the Router Header of each Router Frame for the purpose of detecting Router Frames which are trapped in a loop in the network. When a node receives a Router Frame, the TCNT in the header reports the number of links the frame has traversed. The TCNT is incremented and if it equals or exceeds the TCNT LIMIT, the Router Frame is declared to be undeliverable and diverted for error action. TCNT LIMIT is set to the lesser of 255 or twice the MAXHC. It must be larger than MAXHC for the transient case when a failure of a link or node causes an in-transit frame to back track before finding a surviving route to its destination. Refer to the Operations Interface command TRANSITCOUNTLIMIT.

MAX HOP COUNT (MAXHC)
: When the hop-count (i.e., number of links over a path through the network to a given destination) is greater than or equal to MAXHC, that path is considered to be unusable. Refer to the Operations Interface command MAXHOPCOUNT.

MAX RESISTANCE FACTOR (MAXRF)

When the sum of the resistances over a path through the network to a given destination is greater than or equal to MAXRF, that path is considered to be unusable. Refer to the Operations Interface command MAXRESISTANCEFACTOR.

NODE RESISTANCE FACTOR (NODERF)

The resistance factor of the node for transit traffic. Refer to the Operations Interface command NODERESISTANCEFACTOR.

NETWORK MAX SEGMENT SIZE

The NETWORK MAX SEGMENT SIZE is the MAX SEGMENT SIZE used for all communications in the network, with the possible exception of connections between neighbors via special links, such as Global Memory links. Refer to the Operations Interface command NETWORKMAXSEGMENTSIZE.

ROUTER VALIDATE

If true, the Router will reject an incoming NETCHANGE messages unless its Subject Node Address has been previously identified as a valid node address by an ADD NODE or ADD HOST command. Refer to the Operations Interface command VALIDATE.

ROUTER HEADER SIZE

The size of the Router header in bytes. It is used in the maximum segment size computations. This is a compiled-in constant.

NODE UP TIMEOUT VALUE

When a node first comes up, no NETCHANGE messages are sent out until a timeout period of this length occurs with no LINKCHANGE or NETCHANGE messages received. Refer to the Operations Interface command NODEUPTIMEOUT.

NEIGHBOR RESTART TIMEOUT VALUE

This timeout value is used when a Neighbor is restarting. While this timer runs, the Router does not send NETCHANGE messages about the neighbor. Refer to the Operations Interface command NEIGHBORRESTARTTIMEOUT.

ROUTER MONITOR COPY

Indicates whether a copy of all Router Frames processed is being logged for later analysis. Refer to the Operations Interface command MONITOR.

ROUTER MONITOR SUMMARY

Indicates whether a summary of all traffic passing through the Router is being accumulated and logged. Refer to the Operations Interface command MONITOR (TRAFFIC).

ROUTER MONITOR INTERVAL

At the end of each interval of this length, the traffic profile summary is logged (if ROUTER MONITOR SUMMARY is ON). Refer to the Operations Interface command MONITOR (INTERVAL).

ROUTER VERSION Attributes

The current Version of the Router is defined by three attributes, Router Version ID, Router Version Protocol, and Router Version Compatibility. These attributes are compiled into the software or firmware and exactly define the current Version. Version ID is a string which defines the Router version. Version Protocol is an integer which defines the version of the inter-node Router protocol.

Version Compatibility defines those prior Version Protocols with which this protocol can communicate. This is a 24-bit mask, with the high-order bit corresponding to the protocol level "(Router Version Protocol) - 1", and the low order bit corresponding to "(Router Version Protocol) - 24". A TRUE bit means that the corresponding protocol level is supported by this version of the Router. Refer to the Operations Interface command VERSION.

Version Protocol and Compatibility are included in every Linkchange message. When a node receives this information from a neighbor, it must decide whether or not it can communicate with that neighbor. There are three possibilities:

1. If the received Version Protocol equals the local protocol, communication can proceed at that version.

2. If the received protocol is different from the local protocol, the highest protocol which the two nodes can both handle is used. The compatibility information determines this compatible version.

3. If the two protocols are different and there is no mutually compatible prior version on which the two nodes can agree, the Operations Interface is notified via the NSM and the Router does not consider the other node to be a neighbor for routing purposes. Refer to the Operations Interface report ERROR.

Routing Tables

The update mechanism uses three Routing Tables, Routing Table Info (RTI), Routing Table Current (RTC), and Router Neighbor Table (RNT). RTI is the table of potential routings. There is an entry in the table for each known node. For each node, a number of potential routings are defined, one for each neighbor by which that node can be reached. RTC contains the current routing for each known node. Routing Neighbor Table (RNT) contains information about the logical connection to each neighbor node. Entries in the tables can be accessed or changed by various Operations Interface commands such as ADD NODE, DELETE NODE; ADD STATION, DELETE STATION, MODIFY STATION, NEIGHBOR, HOST, NODE, ROUTINGS, etc.

ROUTING TABLE INFO (RTI)

The RTI is the basic routing table. It is summarized in pictorial form in Figure 5-12. An example of an RTI is shown in this section. The RTI is largely set from information obtained from LINKCHANGE and NETCHANGE messages from neighboring nodes. There is typically an entry in RTI for each node in the network except for the local node. Each entry consists of the Destination Node Address (DNA) and one or more routings to that node, one for each neighbor.

The information contained within the RTI is potentially misleading. It lists ALL POSSIBLE paths to each other node, including some which are illogical. Secondary paths may loop back onto primary paths or may even pass back through the local node. The following list indicates the information stored in the RTI.

Destination Node Address (DNA)
: The node address of the node to which the entry applies.

D-Reachable
: Indicates whether this DNA can be reached through the network. It is the OR of all the R-Reachables.

Routings
: For each DNA, there are typically a number of possible routings, each of which consists of the following ensemble of items: NNA, R-Reachable, Hop Count, and Resistance Factor.

  Neighbor Node Address (NNA)
  : The node address of the neighbor (adjacent) node for this Routing to this DNA. That is, the node address at first hop for this Routing.

  R-Reachable
  : Indicates whether this DNA can be reached through the network via this Routing. When the hop-count over a path through the network to a given destination is greater than or equal to MAX HOP COUNT, or when the sum of the resistances over that path is greater than or equal to MAX RESISTANCE FACTOR, that path is considered to be unreachable.

  Hop Count (HC)
  : The number of hops (links) to the destination via the minimum hop-count path starting with this Routing.

  Resistance Factor (RF)
  : The sum of the resistance factors of links and nodes to the destination via the path of least resistance starting at this neighbor.

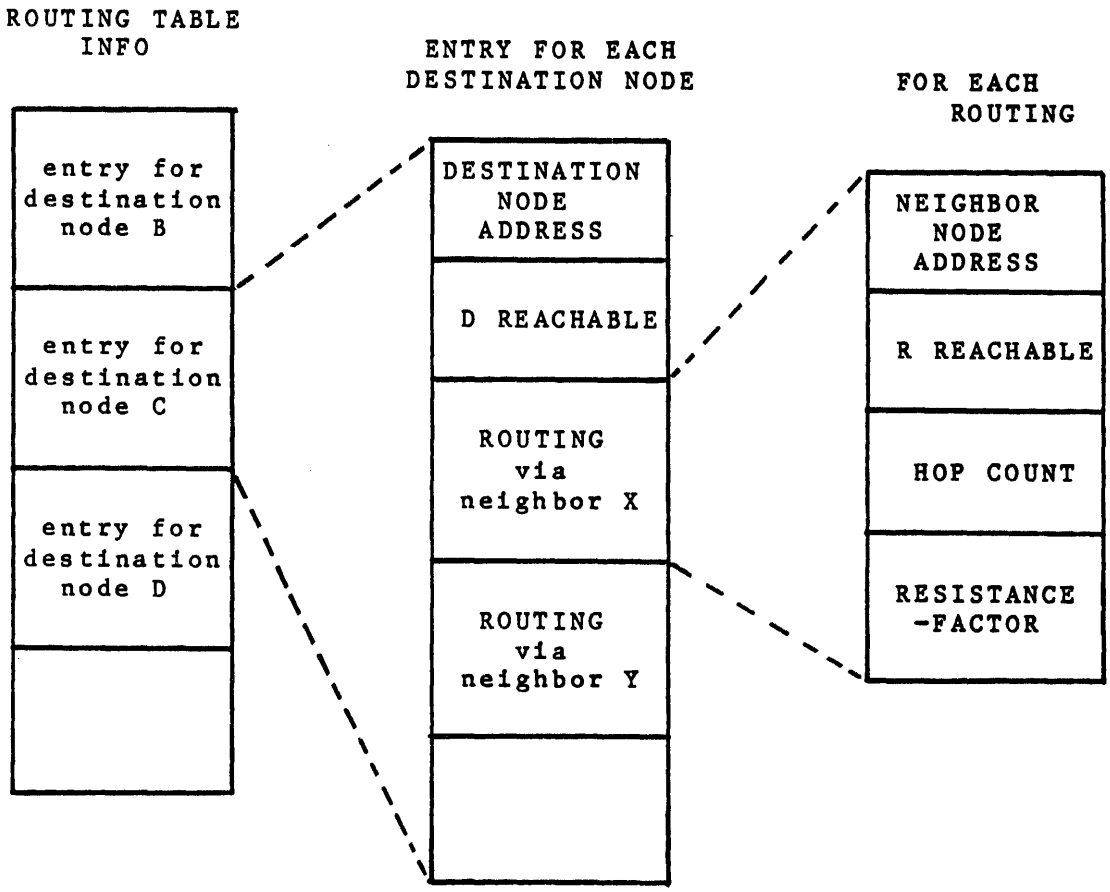


Figure 5-12. Routing Table Info - (RTI)

ROUTING TABLE CURRENT (RTC)

The Routing Table Current (RTC) is the table of Routings which are currently in use. RTC is summarized in pictorial form in Figure 5-13. The RTC is built by the Router based wholly on the contents of the RTI and RNT tables.

There is an entry in RTC for each known node, that is, for each DNA in RTI. Each entry consists of the following ensemble of items: DNA, C-Reachable, NNA, HC, RF, RTC-MSS, RTC-VANID, and Router-STA-Q-ID's. These are described below.

Destination Node Address (DNA)
: The node address of the node to which the entry (i.e., the following current routing) applies.

C-Reachable
: If true, indicates that this node address can be reached through the network. Note that the contents of the remaining items in the Entry are defined only when C-Reachable is true.

Neighbor Node Address (RTC-NNA)
: The node address of the neighbor (adjacent) node for the Routing currently in use to this DNA.

Hop Count (HC)
: The number of hops (links) to the destination via the minimum hop-count path.

Resistance Factor (RF)
: The sum of the resistance factors of links and nodes to the destination via this path of least resistance.

Max Seg Size
: The maximum segment size (in bytes) which can be accommodated through to the destination via this routing. It is equal to the Network MSS except in the special case where neighbor nodes agree to use a larger MSS.

Public Data Network ID
: The identification of the Public Data Network (PDN) currently in use as the next hop to this DNA. Note: zero means that the link is not a PDN. Used to suppress some unnecessary NETCHANGE messages.

Router STA-Q-ID's
: Identify the queues to the station(s) which serve the NNA. There are two queues, Routine and Priority.

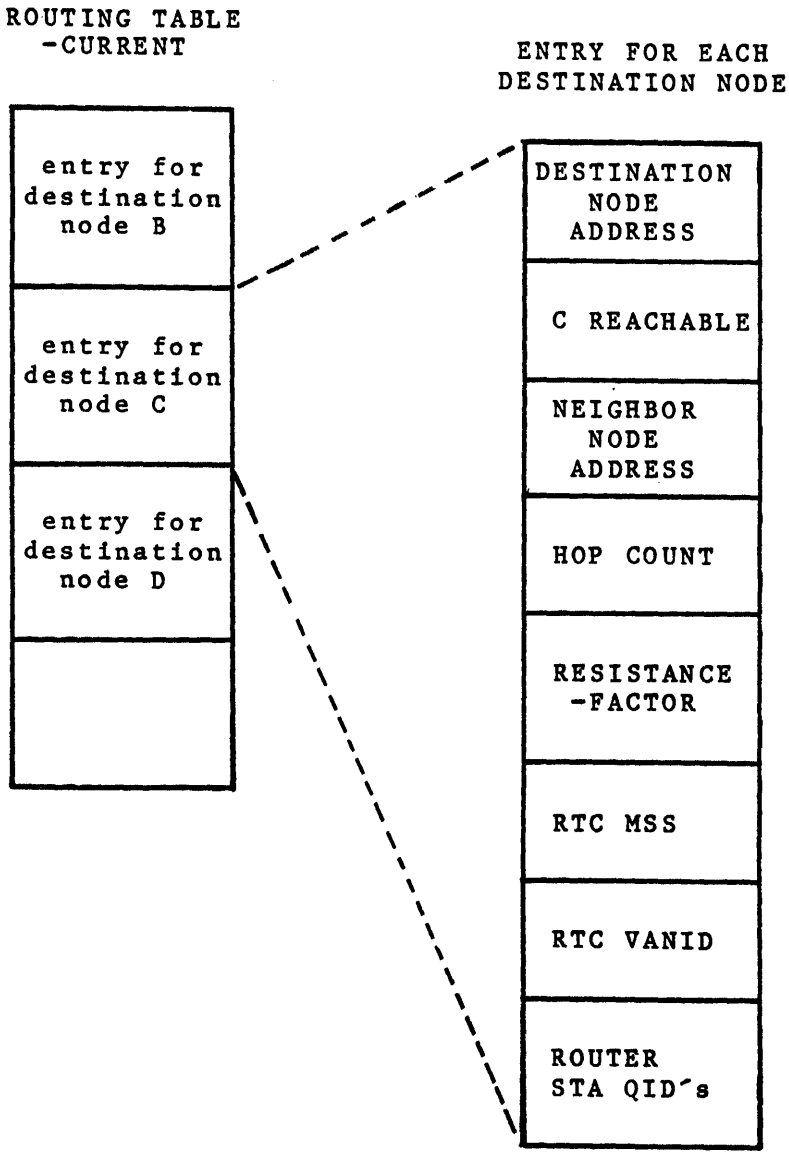


Figure 5-13. Routing Table Current - (RTC)

ROUTER NEIGHBOR TABLE (RNT)

The Router Neighbor Table (RNT) contains detailed information about each node which is currently a neighbor of the local node. There is an entry in RNT for each neighbor node. The RNT is illustrated in Figure 5-14. The components of the RNT are described in the following list.

Neighbor Node Address (NNA)
: The node address of the neighbor to which this RNT entry applies.

Version Working
: Identifies the Version Protocol currently in use with this neighbor.

Link Pending
: Used internally in the routing update action to control the response to ATTACH and DETACH commands and to receive LINKCHANGE messages.

Neighbor Restarting
: Specifies that this neighbor is currently restarting.

Work RF
: The working resistance factor now being used for the logical link to NNA.

Remote RF
: The logical link resistance factor supplied by the neighbor NNA in a LINKCHANGE message.

Local RF
: If the Operations RF below is 0, then this is the logical link resistance factor calculated from the physical link information. If Operations RF is not 0, then the calculated link resistance is overridden and Local RF is equal to Operations RF.

OPNS RF (Operations RF)
: A resistance factor which can be supplied by the Network Services Operations function. If specified, this value overrides the calculated default logical link Resistance Factor.

Maximum Segment Size (MSS)
: The working maximum segment size for traffic to the neighbor, NNA, via this direct logical link. It is the minimum of the MSS's of the physical links which comprise the logical link to NNA. It may be larger than Network MSS, but cannot be smaller.

PDN-ID
: The identification of the Public Data Network by which the NNA is reached. Zero indicates that it is not a PDN. It is determined from the physical link information. If all physical link pdn-id's for this logical link are the same, it is the value for rnt-pdn-id. Otherwise, one of the physical link pdn-id's is arbitrarily chosen.

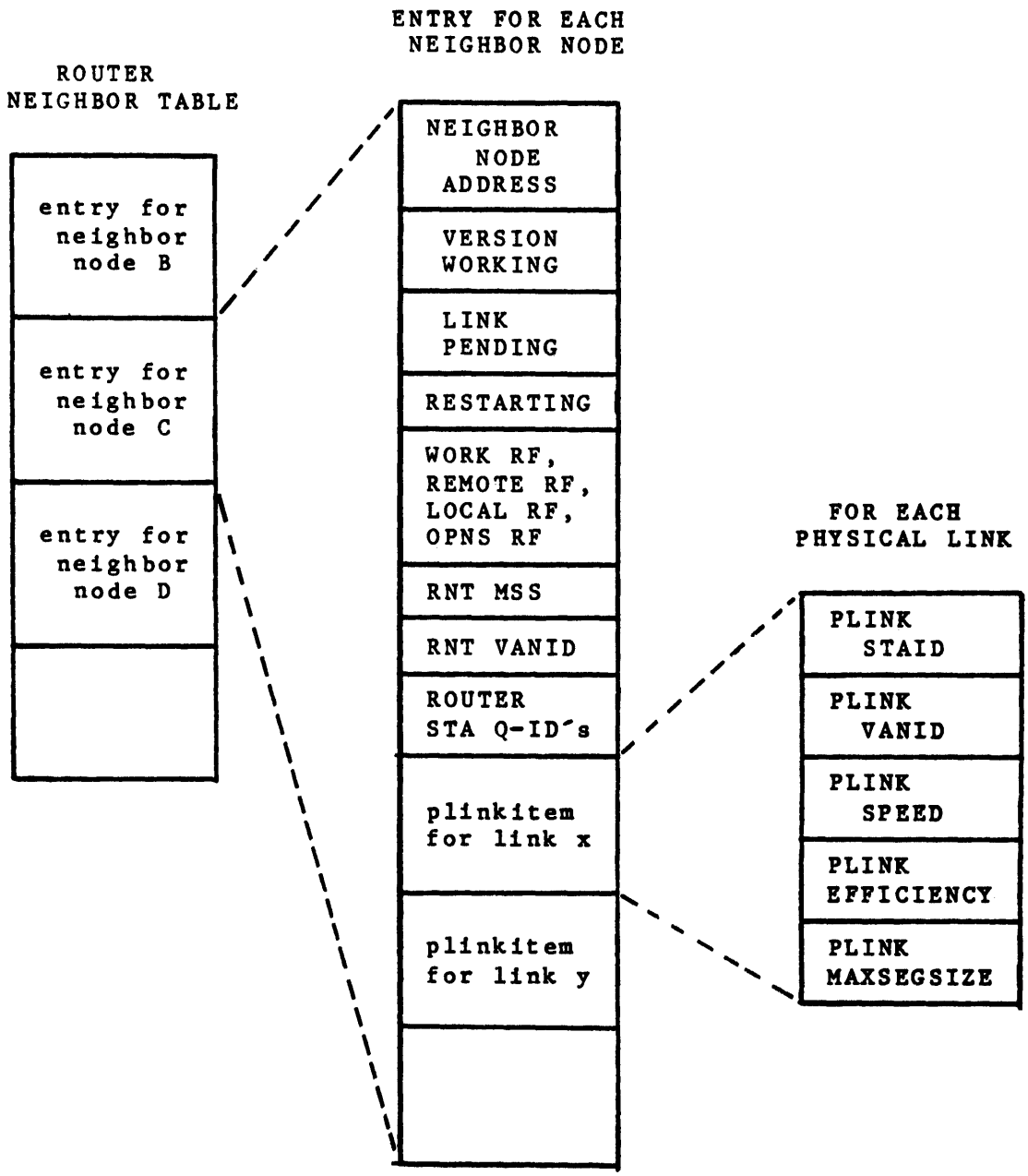


Figure 5-14. Router Neighbor Table - (RNT)

Queue IDs

The identification of the two queues (routine and priority) from the router to the station level for this neighbor node (NNA). In addition, for temporary use during neighbor restart, the two "old" queue id's for this NNA.

Physical Link Items

These physical link items specify the attributes of the physical links to the NNA for the Router. Typically, there is just one such physical link item for a NNA; but for multiple parallel links, there are multiple physical link items, one for each of the parallel physical links. Each Physical Link Item consists of the following sub-items: Physical Link Station ID, Physical Link PDN ID, Physical Link Speed, Physical Link Efficiency, and Physical Link MSS.

Physical Link Station ID

The identification of the local station for this physical link.

Physical Link PDN ID

The identification of the Public Data Network to which this physical link connects. Zero indicates no PDN, that is, a leased or dialed link directly to another node in the network.

Physical Link Speed

The speed of the physical link in bits per second.

Physical Link Efficiency

An estimate of the efficiency of the physical link in percent. Typical values are 95 for full-duplex lines and 45 for half-duplex lines.

Physical Link MSS

The maximum segment size (in bytes) to be used over this physical link for traffic destined for the neighbor.

ROUTER FUNCTIONS

Message Transmission

Three types of frames are processed by the Router's message transmission function: locally originating frames, locally terminating frames, and incoming frames which are to be retransmitted to another node (transit frames).

Locally originating frames are either Router Information Units from the Port Level in the local node, or Router Control Units. A Router Header is generated and appended to these to form a complete Router Frame. The Router Frame is passed to the appropriate local station for transmission to its neighbor node.

The Destination Node Address (DNA) of Router Frames received from stations is checked to determine whether the frame is a locally terminating frame (addressed to the local node address) or destined outside the local node.

The Router Header is stripped from locally terminating frames. Those which are Router Information Units (determined from information in the Router Headers) are queued for the Port Level. Those which are Router Control Units are processed internally in the Router.

After a check of its Transit Count (TCNT), an incoming frame whose DNA is outside the local node (transit traffic) is passed to the appropriate local station for transmission to its neighbor node.

Routing Update

LINK RESISTANCE FACTOR

Each link between two BNA nodes is assigned a link resistance factor (LINKRF) which establishes the relative desirability of using that link for network traffic. High resistance means undesirable; low resistance means desirable. LINKRF refers to a logical link between two nodes in a network. When there are multiple parallel links, LINKRF refers to the composite resistance of the multiple physical links.

A default link resistance is defined for each type of link. It approximates the time for a frame of length MAX SEG SIZE to transit the link (in milliseconds). The equation is:

$$\text{default LINKRF} = \frac{1000 * \text{maxsegsize} * 8}{(\#\text{ of parallel links}) * \text{speed} * (\text{eff}/100)}$$

where: maxsegsize = the network max segment size;
summation is over the multiple parallel links;
speed = speed of the physical link in bits per sec;
eff = an estimate of the transmission efficiency over the link in percent;some suggested values are:
for two way simultaneous, eff = 95;
for two way alternate, eff = 45.

Some sample default link resistance factors are:

maxsegsize =256 bytes	single link		two parallel links	
	eff	eff	eff	eff
link-speed	95	45	95	45
1200	1796	3793	898	1896
9600	225	474	112	237
56000	38	81	19	41

maxsegsize =128 bytes	single link		two parallel links	
	eff	eff	eff	eff
link-speed	95	45	95	45
1200	898	1896	449	948
9600	112	237	56	119
56000	19	41	10	20

NODE RESISTANCE FACTOR (NODERF)

A resistance factor is assigned to each node in the network. Its purpose is to bias the use of each node for the message transiting function. A high resistance factor biases against using the node, low biases toward using it. A default node resistance factor is assigned to each type of node. Thus it is additive with link-resistances, the default node-resistance approximates the time (in milliseconds) for one frame to transit the node (in the absence of any other traffic). Typical default NODERF values are:

B7800 type system = 10
B6800 type system = 30
B4800 type system = TBS
B1000 type system = TBS
B900 type system = TBS

PATH RESISTANCE FACTOR AND HOP COUNT

The sum of the resistance factors of links and nodes over a path through the network is the resistance factor of the path. The resistance factors of the nodes at the ends of the path are not included. The minimum resistance factors are the ones of interest. Route selection is based on using the "paths of least resistance".

Similarly, the number of hops (links) over a particular path through the network from one node to another is called the hop count of that path. Again, the hop counts of particular interest are the minimum hop counts. Hop counts are used only to assist the BIAS mechanism to terminate quickly. They are not used for the route selection process.

Note that between a given pair of nodes, the path of minimum hop count and the path of minimum resistance factor are usually the same, but they need not be.

LINKCHANGE AND NETCHANGE MESSAGES

The Router at each node exchanges routing information with the Routers at neighboring nodes using two types of router control frames. LINKCHANGE messages are used between two Routers when the status of the link (or links) directly connecting the two nodes changes. NETCHANGE messages are used to inform neighboring Routers of the effects of changes in the network topology. The contents and formats of these messages are described in the FRAME FORMATS section.

ROUTING UPDATE MECHANISM

Frames are routed to each destination node, using the path (routing) of least resistance. The resistance of a path is defined as the sum of the resistances of the links and transit nodes in the path. The NODERF's of the nodes at the two ends of the path are not included in the path resistance.

Routing Tables are updated, including re-selection of the in-use routing, when a node comes up or goes down, or when a link comes up or goes down, or when a node or link resistance factor is changed, or when a routing update message (NETCHANGE or LINKCHANGE) is received at a node.

If the minimum resistance factor or the minimum hop count to any destination node address is changed by this update, NETCHANGE messages are sent to neighbor nodes reporting the change. An example of the routing update process is illustrated in Figure 5-16.

ROUTING TABLES - EXAMPLE

Abbreviations used in this section are listed in the following table.

Abbreviations Used in this Section

lna	Local node Address
dna	Destination node address
rea	Reachable (unr - unreachable)
nna	Neighbor node address
hc	Hop Count
rf	Resistance factor
mss	Maximum segment size
pdnid	Public data network ID Note: Zero means that the link is not a PDN
workrf	The working resistance factor now being used for the logical link to NNA
remrf	The logical link resistance factor supplied by the neighbor NNA via a Link Change message.
locrf	Logical link resistance factor calculated from the physical link information
staid	Local station identification
speed	Physical link speed - bits per second
eff	Physical link efficiency in percent
pmss	The maximum segment size to be used over this physical link

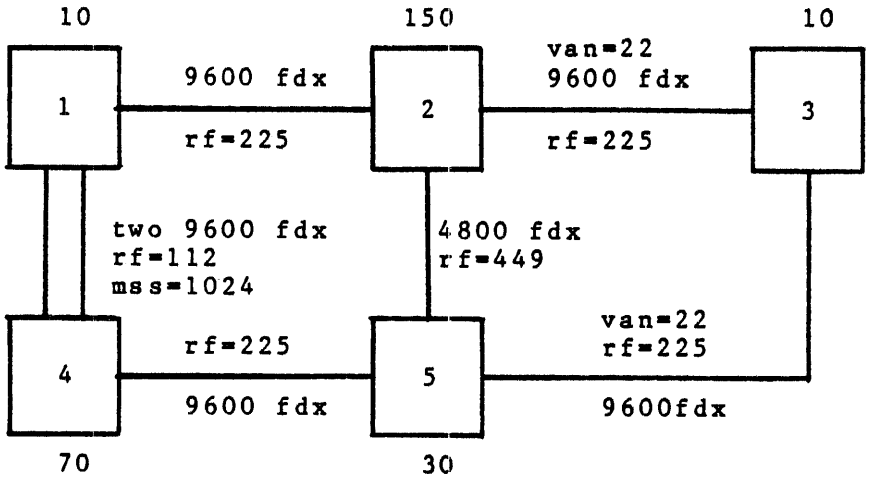


Figure 5-15. Sample Network

The following sample Routing Tables are those which would be found in node 2 of the network in Figure 5-15 above.

```
***** ROUTING TABLE INFO *****
lna dna rea   nna rea hc    rf
  2   1 rea     1 rea  1   225
                3 rea  3   835
                5 rea  3   886
        3 rea   1 rea  3   835
                3 rea  1   225
                5 rea  2   704
        4 rea   1 rea  2   347
                3 rea  3   715
                5 rea  2   704
        5 rea   1 rea  3   642
                3 rea  2   460
                5 rea  1   449
```

```
***** ROUTING TABLE CURRENT *****
lna dna rea nna hc    rf  mss pdnid
  2   1 rea   1  1   225  256     0
          3 rea  3  1   225  256    22
          4 rea  1  2   347  256     0
          5 rea  5  1   449  256     0
```

```
******** ROUTER NEIGHBOR TABLES ********
lna dna workrf remrf locrf   mss pdnid staid pdnid  speed eff pmss
  2   1    225   225   225   256     0    11     0   9600  95  256
      3    225   225   225   256    22    12    22   9600  95  256
      5    449   449   449   256     0    13     0   4800  95  256
```

ROUTING UPDATE EXAMPLE

Refer to Figure 5-15. Assume an initial condition with node 5 not in the network and the network in a stable operating condition. The following is a sequence of Router Control messages exchanged between the Routers in all 5 nodes when node 5 starts.

Msg #	To Node	From Node	Message Type	Subject Node Addr	Hopcount	Resistance Factor
1	5	3	LINKCHANGE			225
2	5	4	LINKCHANGE			225
3	2	5	LINKCHANGE			449
4	3	5	LINKCHANGE			225
5	4	5	LINKCHANGE			225
6	5	3	NETCHANGE	1	2	610
				2	1	235
				4	3	732
				3	0	0
7	5	2	LINKCHANGE			449
8	5	4	NETCHANGE	1	1	182
				2	2	417
				3	3	792
				4	0	0
9	5	2	NETCHANGE	1	1	375
				3	1	375
				4	2	497
				2	0	0
10	3	5	NETCHANGE	1	2	437
				2	1	479
				3	1	255
				4	1	255
				5	0	0
11	4	5	NETCHANGE	1	2	437
				2	1	479
				3	1	255
				4	1	255
				5	0	0
12	2	3	NETCHANGE	4	2	490
				5	1	235
13	5	3	NETCHANGE	4	2	490
				5	1	235
14	2	5	NETCHANGE	1	2	437
				2	1	479
				3	1	255
				4	1	255
				5	0	0
15	1	4	NETCHANGE	3	2	550
				5	1	295
16	5	4	NETCHANGE	3	2	550
				5	1	295
17	2	1	NETCHANGE	5	2	417
18	4	1	NETCHANGE	5	2	417
19	1	2	NETCHANGE	5	2	610
20	3	2	NETCHANGE	5	2	610
21	1	2	NETCHANGE	5	1	599
22	3	2	NETCHANGE	5	1	599

Figure 5-16. Routing Update Example

Messages 1, 2, 3, 4, 5, and 7
Node 5 and all of its neighbors exchange LINKCHANGE messages caused by the topological change.

Messages 6, 8 and 9
Node 5's neighbors each tell node 5 the routes they can provide to all other nodes in the network. Note that none of these routes utilize the new node 5.

Messages 10, 11 and 14
Node 5, acting on the NETCHANGE messages from its neighbors in messages 6, 8, and 9, presents its view of the network to its neighbors.

Messages 12 and 13
Node 3, acting on message 10, advises its neighbors of its new route to node 5, and its improved route (rf=490) to node 4 via node 5. Its previous route to node 4 was via node 2 (rf=732).

Messages 15 and 16
Node 4, acting on message 11, advises its neighbors of its new route to node 5, and its improved route (rf=550) to node 3 via node 5. Its previous route to node 3 was via node 1 (rf=792).

Messages 17 and 18
Node 1, acting on message 15, advises its neighbors of its new route to node 5 via node 4.

Messages 19 and 20
Node 2, acting on message 12, advises its neighbors of its new route to node 5 via node 3 with rf=610.

Messages 21 and 22
Node 2, acting on message 14, advises its neighbors of its new direct route to node 5 with rf=599.

At this point, all of the nodes have advised all of their neighbors of their best routings. None of the last NETCHANGES represent a further improvement to the receiving node over its current routing tables, so the network routing is stabilized.

Router Trace Function

The trace function of the Router allows an operator to determine the path between any two nodes in the network. It is an operations function and, as such, has no effect upon the routing mechanism. The trace is performed by the Router and all results are passed to the originating operator via the NSM. The Router does no analysis of the results of the trace.

ASSOCIATED NODES

There are three nodes associated with the trace function:

1. The initiating (and receiving) node. This is the node which requested the trace and is the recipient of the results of the trace.

2. The trace source node. This is the first node in the trace (the 'from' node).

3. The trace destination node. This is the last node in the trace (the 'to' node).

The initiating node can be any node in the network (including the trace source or trace destination node). The trace source node and the trace destination node cannot be the same.

TRACE RELATED MESSAGES

There are three types of trace related messages: Trace Start, Trace, and Trace Result.

Trace Start

This message is sent by the initiating node to the trace source node if the two nodes are not the same.

Trace

This message is sent by the trace source node to the trace destination node.

Trace Result

This message is sent to the initiating node by the trace source, the trace destination, and any intervening node (i.e., any node which transits the trace message). The Trace Result message contains information about the node that sent it and about the path on which the Trace that generated the Trace Result is forwarded.

TRACE HANDLING

Special handling is given by the Router to the three Trace message types. In the following discussion, routing 'normally' means routing by the path of least resistance, as a Router would route information frames.

Trace Start messages are treated normally by all nodes except the DNA in the Trace Start message (the Trace Source Node). That node sends both a Trace Result to the ONA of the Trace Start (the Trace Receiving Node, the TRNA) and a Trace to the Trace Destination Node. Both the Trace and the Trace Result are routed normally.

When a Trace message is received by any node, a Trace Result message is sent to the TRNA, by the method described in the following paragraph. Then, if possible, the Trace message is forwarded normally to the Trace DNA. If the Trace message cannot be forwarded it is noted in the Trace Result.

When the DNA of a Trace Result is reachable, the Trace Result is routed normally. However, Trace Results have a special routing method when the DNA is unreachable. Because reachability is not transitive (if C is reachable from B, and B is reachable from A, C is not necessarily reachable from A). The Trace Destination Node (as well as any node between the Trace Source and Trace Destination) may consider the Trace initiating node (the DNA of the Trace Result message) to be unreachable. If this occurs, the node routes the Trace Result message to the Trace Source node by the path of least resistance. The DNA of the Trace Result is not changed, it continues to be the TRNA.

Logging

Log entries generated by the Router are placed in the system log at the local node. They report on changes to the Router's tables or other attributes, and on exception conditions.

TABLES AND OTHER ATTRIBUTES

At Network Services initialization, the initial contents of the Routing Tables (the list of known node addresses) and the initial values of the other Router attributes are logged. Subsequent changes to these tables and attributes are also logged.

EXCEPTION CONDITIONS

Reports of errors processed by the Router are logged. When the error concerns a frame, only the Router, Port, and Subport headers of the frame are recorded in the log. The user's text is not logged for security reasons and to reduce the log storage space required.

Monitor

The Router Monitor generates a profile of traffic through the Router. The need for this type of monitoring, as well as the need for the kinds and amounts of information monitored, varies from network to network and from time to time in a given network. Three Router attributes control the Router Monitor; Router Monitor Summary, Router Monitor Copy, and Router Monitor Interval.

If Router Monitor Copy is on, the following data is logged for each Router frame processed by the Router:

- A copy of the Router frame,
- The Router frame length,
- The neighbor node address from which the Router frame was received.

- The neighbor node address to which the Router frame will be sent.

If Router Monitor Summary is on, the log entries are traffic profile summaries accumulated over a time interval defined by Router Monitor Interval.

INTERFACES

Communication with Other Nodes

All communication between a Router and any functions at a remote node is via Router Control Frames to the Router at that node.

Communication within the Local Node

The term "operations" refers to an interface with the operations function at the local node. Nominally, it refers to a human operator, but it can also refer to operations support software which uses files of commands (in place of commands directly entered by an operator). It can also refer to operations support software which may be written to further take over various operator functions. The formats, semantics, and syntax of the operations interface (that is, between the NSM and operations) are in the BNA Reference Manual, Volume 2 (Network Control).

All communication between a Router and its local operations function is via the local Network Services Manager (NSM). One of the functions of the NSM is to convert between Node Addresses, as used in the Router, and Host Names, which are more convenient for a human operator.

All communication between a Router and its local Host Services, and control communication between the Router and the other levels of local Network Services, is also via the local NSM.

The interfaces for message traffic between the Router and the Station Level, and between the Router and the Port Level, can be thought of as queues. (Some individual system implementations use other mechanisms, but they are functionally similar to queues).

There is a single queue from all Stations to the Router. For each neighbor node there are two queues from the Router to the Station Level, one for routine traffic and the other for priority traffic. The basic elements in the queues are Router Frames. Normally, they are Information Frames containing segments of Messages or Responses, but they may also be Network Services Control Frames.

There is a single queue from all Ports to the Router, and there is a single queue from the Router to the Port Level. The basic elements in these queues are Port Frames. Normally, they are Information Frames containing segments of Messages or Responses, but they may also be Port Control Frames.

STATION LEVEL

GENERAL

The Station Level consists of a Station Level Manager (SLM) and one or more Stations which provide communication paths to and from other nodes in the network. The Station Level is the logical entity associated with the end point of a physical communication link. Each Station Level consists of one or more Stations. Each Station is associated with the end-point of a Station Dialog. A BDLC Station Level contains one BDLC Station. An X.25 Station Level can contain several X.25 Stations, each associated with an X.25 logical channel. The logical channels can be associated with either permanent virtual circuits or virtual calls.

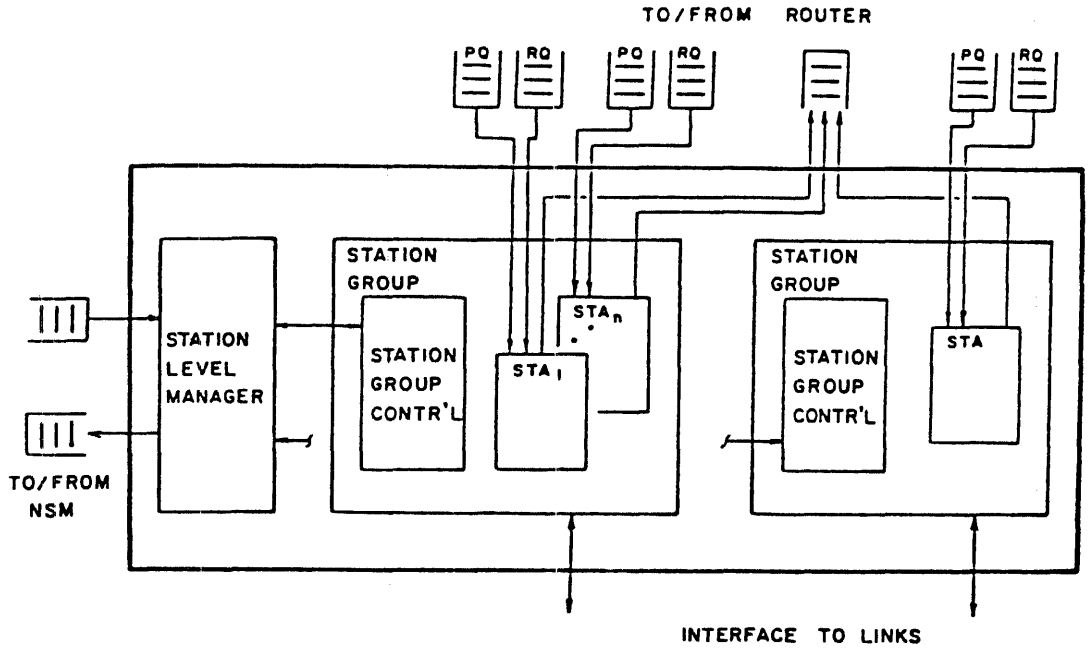


Figure 5-17. Station Level Block Diagram

FUNCTIONS

Station Level Manager Functions

The Station Level Manager performs the following functions within a BNA node:

- Provides for properly coordinated attachment of Stations to the Router.
- Provides optional validation and authentication functions, in which the identity of a potential neighbor (that is, the node at the other end of a potential station connection) is checked before data traffic is

allowed. The validation function checks that the neighbor is included in a pre-stored list of neighbors (the Neighbor Table), thus valid for connection to the local station. This feature is particularly important for switched links. The authentication function checks, using a password mechanism, that the potential neighbor is really the node it claims to be.

- Provides a capability for multiple parallel connections between adjacent nodes.

- Provides an effective capability for the grouping and use of several stations of compatible characteristics when establishing connections.

- Provides an interface for Stations to the Network Services Manager (NSM) and to the Operations Interface via the NSM.

Station Level Functions

DATA TRANSMISSION AND RECEPTION

The primary purpose of the Station Level is to provide error free data transmission between neighbor nodes in a BNA network. Each Station is paired with another Station in another BNA node, either temporarily or permanently, to provide a Station Dialog. The Station Group types, and the Stations contained within them all have the following characteristics:

- The underlying communication media is a point-to-point connection. Multidrop connections are not supported.

- Two way communication is provided by a Station Dialog. This can be either full-duplex or half-duplex depending upon the link characteristics. In BNA, half-duplex is used for BDLC dialed connections; full-duplex is used for BDLC dedicated connections. Station Dialogs can be established between two X.25 Stations. All X.25 Packet Level Station Dialogs within an X.25 Station Group are multiplexed onto a single Link Level dialog.

- A high degree of error detection is provided. For BDLC links, this is provided by a Frame Check Sequence (FCS) field, a 16 bit error detection field in each link frame. Retransmission is automatically provided for frames received in error.

Traffic Flow Priorities

There are two priorities of traffic flow from the Router to the Station Level, priority flow (PQ) and routine flow (RQ). There is only one priority for data flow from the Station Level to the Router. Refer to Figure 5-17.

In addition, Station Level Manager to Station Level Manager traffic is handled separately at a third precedence higher than routine or priority. This SLM to SLM traffic consists of the Station Greeting frames used in Neighbor validation and authentication.

Multiple Parallel Links

Multiple parallel links are established by connecting neighbor systems with more than one pair of stations. The Router is not aware of this capability as segments from the Router are still placed in queues, one pair per neighbor (routine and priority). When stations are configured as multiples to a neighbor, they are all given the same queue IDs. They therefore remove messages from the same set of queues as shown in Figure 5-18. Each station, in effect, is autonomous in that it is not aware that it is acting as one of potentially many multiple parallel links. In the event of errors, messages are placed into the front of the appropriate queues. This allows the resending of the messages as quickly as possible.

The Station Level Manager is the key to the operation of multiple parallel links. The SLM gives the stations the proper queue IDs and maintains a count of active links in parallel. The SLM forwards information through the NSM to the Router so that proper resistance values can be calculated.

Because of varying delays over multiple parallel links, frames can be received at a Router in a different order than they were sent from the transmitting Router. The Port Level is responsible for correctly re-ordering any out-of-order frames.

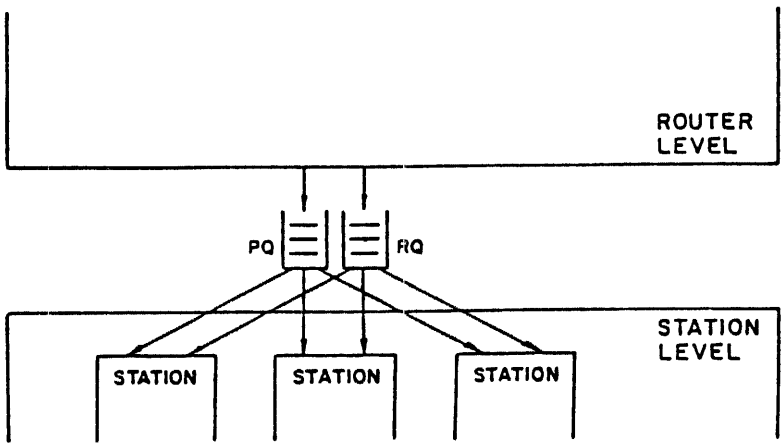


Figure 5-18. Multiple Parallel Links

Station Definition

ENSEMBLES

An Ensemble is a group of stations having certain compatible characteristics. Ensembles allow the user to reference any member of a group of stations rather than specifying a single station with each command.

When making outgoing (BDLC auto-dial) calls, ensembles allow the operator to make a call to a particular neighbor via any available station of an ensemble. This eliminates the need to attempt a call with each station until one is found which is not in use. Similarly, for incoming calls, ensembles allow a node to accept calls over all of the stations of an ensemble, rather than entering an AWAIT CALL command for each station. Incoming ensembles also allow validation on an ensemble basis.

Ensembles, when defined for permanent stations, are used to establish and clear calls by an entire group of stations instead of performing these commands on a station by station basis. Permanent ensembles also allow validation on an ensemble basis.

The default case is the absence of ensembles. The user can perform Station Level operations without defining ensembles. For the situations where ensembles are not wanted, commands are provided to work with individual stations.

Ensembles, if used, are created by use of the ADD ENSEMBLE command. Incoming, outgoing, and permanent ensembles can never have duplicate names. However, one station can be the member of both an incoming and outgoing ensemble. A station that is the member of a permanent ensemble cannot be the member of an incoming or outgoing ensemble. Once an ensemble is created, the ensemble-ID corresponding to the ensemble name is stored in the Station List for each station belonging to the ensemble.

There are restrictions when both ensembles and stations are concurrently defined. A station, if used as a unique (single) station in one direction, cannot be defined as the member of an ensemble in the same direction. This definition applies independently to both the incoming and outgoing directions.

Incoming, outgoing and permanent ensemble-ids are included in both the Station List and Neighbor Table.

Adding an ensemble (ADD ENSEMBLE command) causes stations to be grouped together, and subsequently referenced collectively. A common use for an ensemble is in the case of auto-answer lines, where no particular line is dedicated to any one user. In this case it is convenient to allow a user to come into the system on any one of the stations in the group (ensemble).

PROFILES

A Profile is a collection of Station attribute names and values. It is defined independently of a Station, to be used later as a source of attribute values when Stations are added to the node's network configuration.

Deleting a profile (and subsequently redefining it) has no effect on a Station which was added using that profile. The profile is not a Station attribute. It is only a source of attribute values at the time a Station is added.

The Operations Interface configurational command ADD STATION defines a station and makes it available for use in preparation for call establishment. The ADD PROFILE and MODIFY STATION commands can also be used to define the station. Refer to Figure 5-19.

Adding a connection (ADD CONNECTION command) identifies neighbors for later use in the station validation process, and supplies further information required for automatic call initiation. Refer to Figure 5-19.

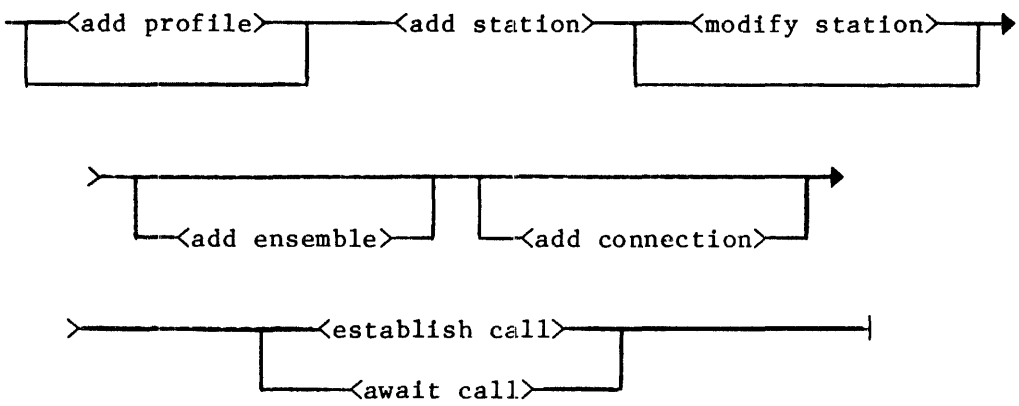


Figure 5-19. Relationship of Call Establishment Commands

Call Establishment

The commands ESTABLISH CALL and AWAIT CALL start the call establishment process.

The ESTABLISH CALL command causes the physical connection process to begin. For auto-dial this is the dialing of the call. For a permanent type connection, it is the starting of station dialogs to a neighbor.

Awaiting a call (AWAIT CALL command) is similar to establishing a call except that the station is put into a receive state. For auto-answer, this is waiting for the telephone line to start ringing.

Both commands result in the initialization of BDLC protocol, station level validation and authentication, and the reporting of the new link to the Router.

The call establishment process includes the following (refer to the description of commands in the Station Group sub-section):

1. Selection of a station.
   The station (or ensemble) is specified in the ESTABLISH CALL or AWAIT CALL command.
2. Establishment of the physical connection.
   This is the automatic or manual dialing or answering of the call and Opening the Connection Port Dialog. For X.25 connections, it includes the establishment of all underlying connections, notably the opening of the link level station.
3. Establishment of the link level connection.
   For BDLC stations, this is opening the Station Dialog. For X.25 Logical channels this is opening the X.25 link level Station Dialog, including validation of any parameters supplied by the common carrier as part of the connection establishment process.
4. Station level neighbor node validation and authentication.
   These are performed using station greetings messages. If successful, the values in the station list are updated.
5. Attach the station to the Router.
   The station is attached to the Router and the Router is advised with an ATTACH message via the NSM. This results in activation of the Router update mechanism.

Call Clearing

This can be initiated in one of three ways:

1. By an Operations Interface CLEAR CALL command,

2. By a coordinated link level clearing exchange (DISC-UA exchange in BDLC),

3. By an unexpected disconnection detected by the Station Group.

The following steps occur:

1. The Station Dialog is closed. No further frames are taken from the Router queues, and any unacknowledged frames are placed back on those queues. Incoming traffic is stopped. If the SAVE indicator is true, the station remains SAVED.

2. The physical connection is closed, except for X.25 packet level stations. For BDLC switched stations, the Connection Port is reopened if AUTO-INIT is true and the SAVE indicator is false.

3. The station list and neighbor tables are updated and the Router is advised with a DETACH message, causing activation of the Router update mechanism.

Access Control

NEIGHBOR NODE VALIDATION

The neighbor node validation function permits the Operations function at a node to define which nodes are acceptable neighbors. The function is performed by comparing the origin node address in received greeting messages against the preloaded neighbor table, which defines the valid neighbors for each station or ensemble of stations.

There are two classes of neighbor node validation performed within the Station Level:

1. Validation after the station dialog is opened, before the station is attached to the Router.

2. Validation performed during the establishment of a station dialog, for example, validating parameters in an X.25 Incoming Call Packet.

NODE AUTHENTICATION

The node authentication function verifies that a neighbor node really is the node it claims to be. The function is performed by comparing the password in a received Greeting 1 with a pre-stored password for the node identified by the origin node address in Greeting 0 and 1.

The SAVE Function

The SAVE command allows takeover of the station for test purposes by preventing it from being re-opened when it is closed. No connections can be established with the station using any command from the SLM such as the ESTABLISH CALL or AWAIT CALL commands. The SAVE command can be applied to any BDLC station or to an X.25 Station Group. The save action takes place when the station becomes closed with a CLEAR CALL command, remote closing, or link failure.

The READY command negates the effect of the SAVE command. The READY and DELETE commands are the only allowable commands to a saved station. Refer to the BNA Reference Manual Volume 2 (Network Control).

Confidence and Diagnostic

Several commands, called Station Manual commands, are provided to support confidence and diagnostic testing of BDLC stations. They allow an external agent to sequence the station through the various steps involved in establishing or clearing a connection, as an aid to troubleshooting. They are used as maintenance operations to determine the condition of a station and its link level operation. Refer to the BDLC Station Group sub-section for further information on these commands.

These commands contain parameters which identify the station and the action to be performed by the station. The commands are rejected if the SLM phase is incorrect, if the station does not exist, if an automatic command is in process on the station, or if they attempt to cause an illogical sequence of events.

The Open Connection Port Dialog (OCPD) command is allowed on any closed station, and in addition to opening the connection port to allow the transmission of test messages, sets MANUAL COMMAND MODE to TRUE to prevent any normal use of the station. No BDLC protocol is available.

When the Connection Port Dialog is open, Close Connection Port Dialog (CCPD), Open Station Dialog (OSD), Open X.25 Station Dialog (OXSD), Close X.25 Station Dialog (CXSD), and Send Test (ST) can be used together with the Inquiry commands to determine the condition of the station and hardware connection. Opening a station dialog initializes BDLC protocol. Note that the OSD command must precede an OXSD command.

When the station dialog is open, a VALIDATE AND ATTACH command starts the Greetings interchange process, which attaches the station to the Router and takes it out of manual command mode. If a VALIDATE AND ATTACH command has not been entered for a station and a Greeting 0 is received by the SLM, the Greeting 0 is saved for use when the VALIDATE AND ATTACH command is entered. When the VALIDATE AND ATTACH command is entered, the Greetings exchange process is initiated and the station sends Greeting 0. The Greetings exchange process continues normally from this point. When Neighbor Table entries are built for Stations in Manual command mode, temporary connection type entries are built using the information contained in the Greeting 0 and Greeting 1 messages, unless neighbor table entries already existed.

A call can also be cleared manually. A DETACH command causes the same steps (for an ATTACHED Station) as a CLEAR CALL command, except that the connection port dialog is not closed. It severs the connection between the station and the router, and leaves the station in a condition that permits further manual operations. The station report to the router causes the station's link to be removed from any routing.

SEND TEST causes a test message to be sent to the neighbor station. The connection port dialog must be open.

When the connection port dialog is CLOSED, the station goes out of manual command mode.

REPORTS

The following reports are sent to the NSM. They may generate entries to the system log (controlled by the NSM's LOGGING option), and may be displayed on the operator's console (controlled by the REPORTS command).

Attach Report
: Indicates a new connection.

Detach Report
: Indicates a severed connection.

Link Reset Remotely Report
: Indicates that a remote station has opened or reopened a Station Dialog, rejected a frame, or sent an unexpected Disconnect Mode or Unnumbered Acknowledgment frame to the local station.

Neighbor Restart Report
: Indicates that GREETING 1 has been received with INITIALIZING-INDICATOR equal INITIALIZING and the neighbor was ATTACHED.

Neighbor Remote Busy Report
: Indicates that all connections to the neighbor are in a remote busy state (longer than the value of the NEIGHBOR BUSY TIMEOUT attribute). Operator action is required to resolve this situation.

Validation Failure Report
: Indicates a failure of the Station Level Validation, either on this end or the other end of the connection.

BDLC Test Command Received Report
: Indicates that a BDLC TEST Frame was sent by a remote BDLC Station Group as a command, and presents the I-field of the frame, if any. The local BDLC Station Group returns a Test Received Report which the SLM formats into this report.

BDLC Test Response Received Report
: Indicates a BDLC TEST Frame was sent by a remote BDLC Station Group as a response to a local Station Group's BDLC TEST Command and presents the I field of the frame, if any. The BDLC Station Group returns a positive response to an SLM SEND TEST command which the SLM formats into this report.

Station Failure Report
: Indicates Station Group failure in completing a specified command to which the SLM has already returned an Operations Interface response. This report is used for SEND TEST commands (after the SLM sends the frame to the Station Group), AWAIT CALL commands (after the SLM has issued the OPEN CONNECTION PORT DIALOG command), and ESTABLISH CALL commands for permanent stations or ensembles (after the local Station Group has reached any pending condition which requires action by the remote station).

Station Level Monitor Report
Indicates sampling of the Station Group Counters as a result of the Monitor Indicator and Monitor Interval functions.

Station Log Report
The SLM adds identifying information to any reports received from the Stations, tags them as to whether disconnection was caused or not, and sends them to the NSM as a STATION LOG REPORT. Refer to Section 8, Logging and Monitoring, for details.

STATION LEVEL ATTRIBUTES

The following attributes are identified at the Station Level:

LOCAL NODE ADDRESS
Specifies the node address of the local node.

STATION LEVEL VERSION
Defines the current level of the SLM software. It consists of three subfields: Protocol, Compatibility and Version ID. Station Level Version determines protocol compatibility of the SLM during the Greetings interchange. These attributes allow various versions to operate in the same network without any action by the operators at the various nodes.

Version Protocol
Identifies present protocol level of SLM.

Version Compatibility
Specifies the prior protocol levels that the SLM supports. The first bit (MSB) identifies the version prior to the present one, the second most significant bit identifies the version prior to that, etc. If a bit = ONE (TRUE), the SLM supports the identified protocol version.

Version ID

NETWORK VERSION
This field is used by a network administrator. All nodes in a network must have the same value in NETWORK VERSION. This attribute is included in outgoing Greeting 1 messages, and the value in incoming Greeting 1 messages is compared to this attribute. An exact match is required for the connection to be accepted, even if zero length. Null matches null.

NETWORK MAX SEGMENT SIZE (NMSS)
Specifies the default max segment size for this network.

GREETING TIMEOUT
Specifies the amount of time to wait for the Station Level Greetings interchanges. Failure to accomplish all exchanges in the time allotted results in a validation failure.

NEIGHBOR BUSY TIMEOUT
Specifies the time that all links to a neighbor must be BUSY before a report is sent to the NSM.

ALL NEIGHBOR VALIDATE (ANV)
Specifies whether or not to check the identity of every remote node during the Station Level validation process (Greetings interchange). If ANV is false, neighbor validation is performed only on those stations which have an ADD CONNECTION command associated with them, that is, Neighbor Validate is true. The remote node's address is always checked on outgoing calls regardless of ANV or individual station Neighbor Validate values. This is the "wrong number" check.

MONITOR INDICATOR
Indicates that one or more stations have monitor counters set to count and are sampled at intervals determined by MONITOR INTERVAL.

MONITOR INTERVAL
See MONITOR INDICATOR.

Station List

The Station List keeps data about the stations and the Station Groups in the node, and has one entry for each station. For BDLC Station Groups, there is one station in each Station Group, so there is one station entry in the list for each Station Group. For each X.25 Station Group there is an entry for the Link Level (HDLC or BDLC) Station, plus an entry for each of the Packet Level Stations. All information relative to the status of each communication channel, either logical (X.25 LCN) or physical, is stored in the appropriate entry in the Station List. Refer to Figure 5-20.

Within each station entry in the Station List are the following items:

STA ID
Identifies the station.

HARDWARE ID
Relates the logical station described by this entry to the actual physical station (line adapter). It has no meaning for X.25 Packet Level Stations.

STATION TYPE

Values: BDLC DEDICATED
BDLC SWITCHED (with ACU capability)
BDLC SWITCHED (Auto answer with manual dial capability)
X.25 Switched Virtual Circuit (SVC)
X.25 Permanent Virtual Circuit (PVC)
X.25 Link Level

LINK LEVEL STATION ID

This attribute is only used in X.25 Packet Level station entries. It identifies the link level station of the Station Group of which this station is a member.

X.25 LCN (LOGICAL CHANNEL NUMBER)

This item is only valid if Station Type = X.25 SVC or PVC. The LCN is used in command response interchanges between the SLM and the X.25 Station Group. The LCNs are assigned by the Station Group, not by the SLM. For PVCs, the LCN is permanently associated with a particular station ID when the ADD X.25 STATION command is entered.

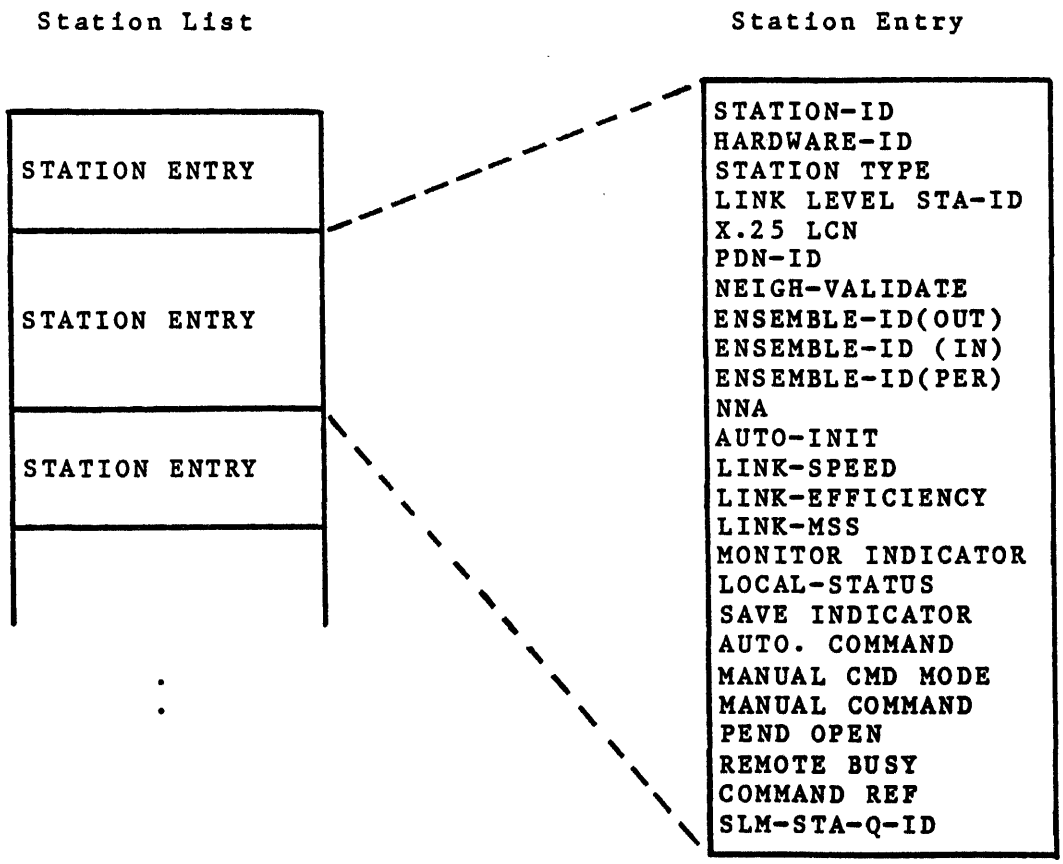


Figure 5-20. Station List

PDN ID

The identification of the Public Data Network to which this station is or will be connected. A null PDN ID indicates that it is not connected to a PDN. Since more than one X.25 Station Group can be connected to the same PDN, this item is not redundant with Station Group ID. Packet Level X.25 Stations default to the value in the related Link Level Station entry.

NEIGHBOR VALIDATE

Specifies verification of the remote node address during the Station Level validation process (Greeting interchange). The remote node's address is always checked on outgoing calls.

OUTGOING ENSEMBLE ID

Used in the process of selecting a station when making an outgoing connection. A value of NULL means that the station is not the member of an outgoing ensemble. This prevents station selection unless this station is explicitly referenced. This item is only allowed for BDLC switched stations with ACU capability and for X.25 Switched Virtual Circuits (SVCs).

INCOMING ENSEMBLE ID

Only stations that can receive incoming calls, namely BDLC SWITCHED and X.25 SVCs, have a (meaningful) Incoming Ensemble ID. A value of NULL in this field indicates that this station is not the member of an incoming ensemble and must be referenced explicitly.

PERMANENT ENSEMBLE ID

Used to select permanent stations when establishing or clearing several permanent connections with one command. A value of NULL indicates that the station is not a member of a permanent ensemble and must be explicitly referenced. This item is allowed only for BDLC DEDICATED stations and X.25 PVCs.

NEIGHBOR NODE ADDRESS (NNA)

Used in reports of changes in the connectivity to the NSM and Router. The NNA is also set in each open station and is included with each Link Information Unit (LIU) that is sent to the Router.

AUTO INIT

Used during initialization and when performing certain operational commands. For BDLC dedicated stations, TRUE indicates that an attempt to Open the Station Dialog is made during initialization. For BDLC SWITCHED stations, TRUE indicates that the "normal" or "initial" state of the station is AWAIT CALL, rather than closed.

LINK SPEED AND LINK EFFICIENCY

These parameters are stored in the STATION LEVEL and are sent to the ROUTER in ATTACH messages. They are used to determine the resistance factor of the link. The units are: for SPEED, bits per second; for EFFICIENCY (EFF), percent.

LINK MAX SEGMENT SIZE (LMSS)

Specifies the size of the largest LIU that can be transmitted (or received) over this link. If the value is NULL, the NETWORK MAX SEGMENT SIZE (NMSS) is used. LMSS is used during the STATION LEVEL validation to determine the WORKING LMSS (WLMSS) for the link. This value can be changed by the SET SLM ATTRIBUTES command when the station is closed, and must be greater than or equal to the NMSS attribute.

WORKING LINK MAX SEGMENT SIZE (WLMSS)

Established during Station Level validation and sent to the ROUTER in the ATTACH message.

MONITOR INDICATOR

Indicates that the monitor counters in the Station Group are active. When first set, the SLM must activate the monitor counters in the Station Group. When a command to close the station is given, the residual count in the counters is sent to the NSM if the value of this item if TRUE.

LOCAL STATUS

Maintains the status of the station and controls the sequencing of the steps involved in establishing or clearing a call.

AUTOMATIC COMMAND

Specifies the operation presently being performed by the SLM with respect to this station.

Values: NULL
ESTABLISH CALL
CLEAR CALL
AWAIT CALL

MANUAL COMMAND MODE

Indicates that the station is being controlled manually. Used for maintenance operations only.

MANUAL COMMAND

Specifies the manual operation currently in progress, if any.

Values: NULL
OCPD
OCPD (Call Data)
CCPD
OSD
CSD
OXSD
CXSD
VALIDATE AND ATTACH
DETACH
SEND TEST

PEND OPEN
Indicates that the outcome of an opening action is still to be determined and is used in Initialization Phase 3.

REMOTE BUSY STATUS
Indicates whether the remote station is busy or not, based upon the last received link level flow control frame.

COMMAND REFERENCE
Retains the reference field of the command in progress (relative to this station) to be returned in the response to the command.

SAVE INDICATOR
Indicates that this BDLC or X.25 link level station is to be saved, or prevented from re-opening when it becomes closed. Refer to the Operations Interface command SAVE.

Neighbor Table

The Neighbor Table (NT) contains information about active and potential neighbors. Refer to Figure 5-21.

The NT contains one neighbor entry for each (incoming) neighbor with which this node can communicate. There is a neighbor entry for each neighbor for whom Calling Data is kept (the NT serves as the telephone number directory). In addition, there is an entry for each neighbor with which there is active communication (attached), even if no ADD CONNECTION command was entered. All of the stations that are currently connected to this neighbor are in the STATION ID entries in the connection type entries of this table.

Each neighbor entry consists of a header and one or more connection type entries.

NEIGHBOR ENTRY HEADER

Neighbor Node Address
Identifies the neighbor.

Version Working
Identifies the Version Protocol level currently in use with this neighbor.

My Password To Remote SLM
Refer to Section 7, Access Control.

Remote SLM Password To Me
Refer to Section 7, Access Control.

Neighbor Status

Maintains the status CLOSED, INITIALIZING, or ATTACHED of the neighbor as known at this node. Controls the resetting of links to the neighbor when it is recovering.

Neighbor Remote Busy Status

Indicates the logical flow control status NOT BUSY, WAITING, BUSY of the neighbor. It is BUSY when all the stations ATTACHED to the neighbor have REMOTE BUSY STATUS = BUSY.

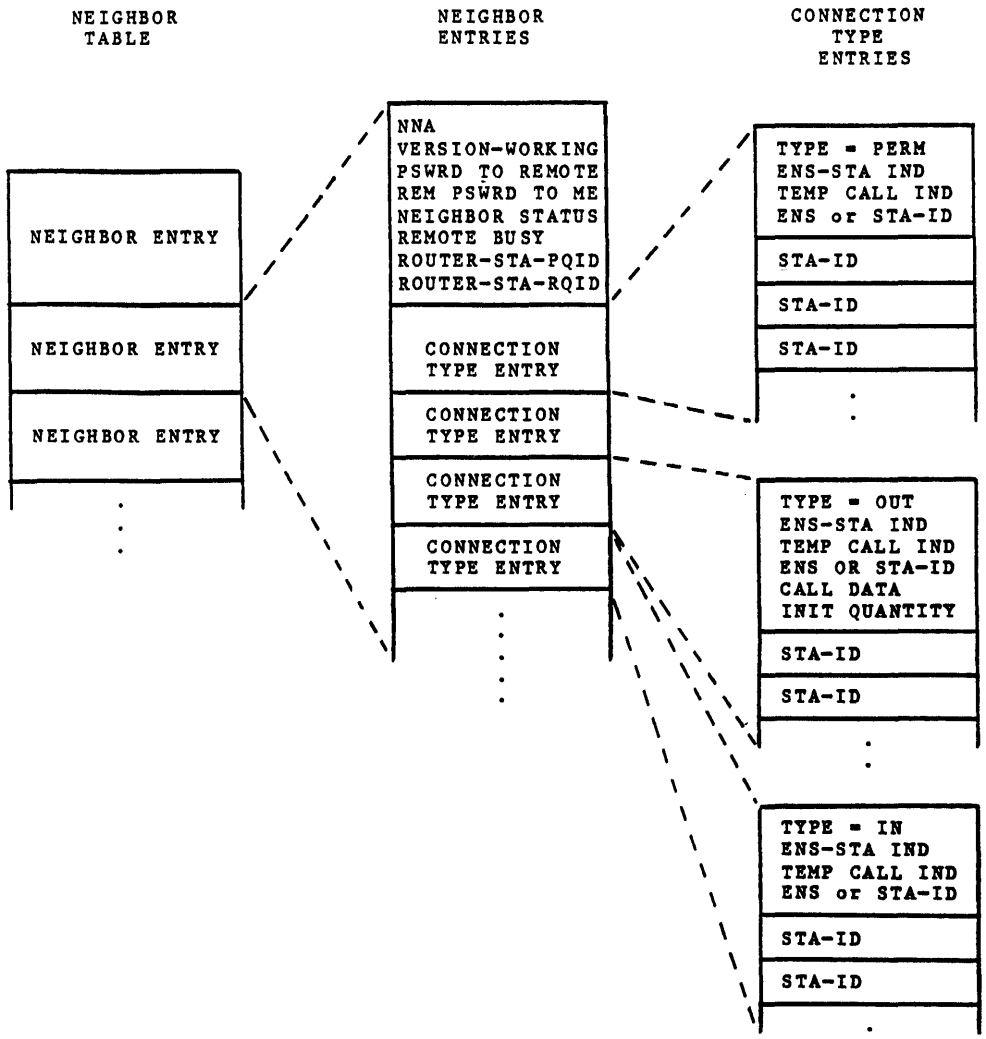


Figure 5-21. Neighbor Table

PERMANENT CONNECTION TYPE ENTRY

There is one PERMANENT CONNECTION TYPE ENTRY for each ensemble or each station (when no permanent ensemble containing this station is defined) which can be used to establish a permanent connection, either BDLC DEDICATED connection or X.25 PVC. Each entry consists of a header and possibly a Station Identifier list. The entry consists of:

Type = PERM

Ensemble or Single Station Indicator

Indicates whether this connection type entry refers to one station or a group of stations (an ensemble). If this item specifies one station, the ensemble identifier entry within this connection type entry is null. If this item specifies an ensemble, the ensemble identifier entry within this connection type entry is not null.

Temporary Call Indicator

Indicates whether this connection type entry is to be removed from the neighbor table after the call is cleared. If the connection type entry is provided by an ESTABLISH CALL command, the indicator is TRUE and the connection type entry is deleted after the call is cleared. If the connection type entry is provided by an ADD CONNECTION command, the indicator is FALSE.

Ensemble Identifier Number or Station Identifier Number

Station Identifier list

If this entry specifies an ensemble.

OUTGOING CONNECTION TYPE ENTRY

There is one OUTGOING CONNECTION TYPE ENTRY for each ensemble or each station (when no outgoing ensemble containing this station is defined) that can be used to call this neighbor via dial, manual dial, or X.25 SVC capability. OUTGOING CONNECTION TYPE ENTRIES have a header plus STATION IDENTIFIER entries in a STATION IDENTIFIER list that are created dynamically as Stations are ATTACHED.

Type = OUT

Ensemble or Single Station Indicator

Same as described for PERM above.

Temporary Call Indicator

Indicates whether this connection type entry is to be removed from the neighbor table after the call is cleared. If the connection type entry is provided by an ESTABLISH CALL command, the indicator is TRUE and the connection type entry is deleted after the call is cleared. If the connection type entry is provided by an ADD CONNECTION command, the indicator is FALSE.

Ensemble Identifier Number or Station Identifier Number
Identifies the ensemble identifier or station identifier of the station(s) now connected to this neighbor or that can be used to call this neighbor.

Call Data
This item, whose coding and value are both implementation dependent and provided by the common carrier, is used to establish the connection. For BDLC SWITCHED with ACU, it consists of the telephone number (in a form suitable for the connection port and ACU). For BDLC SWITCHED without ACU, no Call-Data is contained in the entry. For an X.25 SVC, it includes at least the PDN X.25 address of the neighbor and Closed User Group information.

Init Quantity
Specifies the number of connections that are to be automatically initiated to this neighbor at initialization (via this ensemble or single station).

INCOMING CONNECTION TYPE ENTRY
There is one INCOMING CONNECTION TYPE ENTRY for each ensemble or each station (when no incoming ensemble containing this station is defined) over which incoming connections are accepted from this neighbor. Each INCOMING CONNECTION TYPE ENTRY consists of a header and a number of STATION IDENTIFIER entries that are dynamically created as incoming connections are established and ATTACHED.

The entry consists of:

Type = IN
Same as PERM.

Ensemble or Single Station Indicator
Same as described for PERM.

Ensemble Identifier Number or Station Identifier Number
Used to determine the acceptability of incoming connections by the incoming station or ensemble. For example, this can be used to allow incoming connections from a particular neighbor via BDLC SWITCHED lines, but not via X.25 SVCs.

Profile Table

This table contains an entry for each profile. Each entry contains the Profile name and a list of attribute name, attribute value pairs. Its use is described in Section 6, Initialization. Refer to the Operations Interface command ADD STATION.

A profile for the X.25 link level station includes link level attributes but not those for Packet Level Stations.

STATION LEVEL INITIALIZATION

Station Level Initialization is part of node Initialization, which is started by a NET+ command or by Host Initialization when the system starts operating. Refer to Section 6 - Initialization. Station Level Initialization proceeds in four distinct phases, coordinated with Initialization operations in other levels by the NSM.

Phase 1: Values from the Initialization file set the SLM attributes LOCAL NODE ADDRESS, NETWORK VERSION, NETWORK MAX SEGMENT SIZE, GREETING TIMEOUT, REMOTE BUSY TIMEOUT and ALL NEIGHBOR VALIDATE.

Phase 2: The Station List, Neighbor Table, and Profile Table, and the Station Groups themselves are built from Operations Interface configurational commands in the Initialization file. These commands are ADD PROFILE, ADD STATION, ADD ENSEMBLE and ADD CONNECTION. Associations are established between the station groups and their associated hardware.

Phase 3-1: Opening of permanent connections and recovery of pre-existing dialed connections are initiated.

Phase 3-2: Responses from actions initiated in Phase 3-1 are processed. Successful responses cause the continuation of the opening or call establishment process through Station Level Validation and results in an ATTACH message to the Router.

Phase 3-3: Outgoing calls specified to be made at Initialization are initiated. For each entry in the Neighbor Table with type=OUT, enough ESTABLISH CALL commands are generated to bring the total number of call establishment attempts up to the value specified by INIT QUANTITY.

Phase 3-4: Responses from actions initiated in Phase 3-3 are processed.

Phase 4: Initialization Complete Control Frames signifying Station Level Initialization complete are sent to neighbors via all OPEN stations. This causes them to set Neighbor Stations to ATTACHED in their neighbor table entries for this node. All BDLC switched stations and X.25 SVC's whose AUTO INIT attribute is TRUE are placed in an await call state. The SLM goes to a normal operation state and the NSM is notified.

BDLC STATION GROUP

General

The BDLC Station Group contains a single station, called a BDLC Station. It provides a balanced, point-to-point BDLC protocol for the link level. Both stations have identical data transfer and link control capability; that is, there is no Primary/Secondary relationship.

Two hosts in a BNA network are interconnected using a STATION in each host and a data communications link, a telephone line with data sets having an RS232C interface between each STATION. The STATION supports the physical connection and associated connection protocols of a host to a single data communications link. The connection protocols are:

RS232/V.24 for

- Dedicated lines.
- Manual or auto dial-in of switched lines.
- A Manual dial of switched lines.

RS366/V.25 for

- Auto dial-out of switched lines.

Refer to Figure 5-22 - Communication Line Interface - RS232C/RS366

The BDLC station supports the following:

- Two way simultaneous operation over dedicated, full-duplex (FDX) lines.
- Two way alternate operation over switched, half-duplex (HDX) lines.

A pair of BDLC stations interconnected with a communication line provide two systems called a Connection Port Dialog and a Station Dialog.

CONNECTION PORT DIALOG (CPD)

The CONNECTION PORT DIALOG is a system consisting of

- a pair of CONNECTION PORTS and
- a single communication line between them.

DESIGNATION USED IN CONN.-PORT	SIGNAL NAME	CONNECT PIN NUMBER	RS232	CCITT V24/V28	
	PROTECTIVE GROUND	1	AA	101	DATA SET (MODEM)
	SIGNAL GROUND	7	AB	102	
TD	TRANSMITTED DATA	2	BA	103	
RD	RECEIVED DATA	3	BB	104	
RTS	REQUEST TO SEND	4	CA	105	
CTS	CLEAR TO SEND	5	CB	106	
DSR	DATA SET READY	6	CC	107	
DTR	DATA TERMINAL READY	20	CD	108.2	
RI	RING INDICATOR	22	CE	125	
DCD	RECEIVE LINE SIGNAL DETECTOR	8	CF	109	
CH	DATA SIGNAL RATE SELECTOR (DTE SOURCE)	23	CH	111	
TC	TRANSMITTED SIGNAL ELEMENT TIMING	15	DB	114	
RC	RECEIVED SIGNAL ELEMENT TIMING	17	DD	115	
SS	SELECT STANDBY	(1)		116	

(1) PIN NUMBER IN ACCORDANCE WITH TELEPHONE COMPANY SPECIFICATION, E.G., UK USES PIN 24.

		CONNECT PIN NUMBER	RS-366	CCITT V25	
	PROTECTIVE GROUND	1	AA	212	AUTOMATIC CALL UNIT. (ACU)
	SIGNAL GROUND	7	AB	201	
CRQ	CALL REQUEST	4	CRQ	202	
DLO	DATA LINE OCCUPIED	22	DLO	203	
(1) COS	CALL ORIGINATION STATUS	13	COS	204	
ACR	ABANDON CALL AND RETRY	3	ACR	205	
PND	PRESENT NEXT DIGIT	5	PND	210	
DPR	DIGIT PRESENT	2	DPR	211	
NB1	LOW ORDER BINARY DIGIT	14	NB1	206	
NB2	SECOND ORDER BINARY DIGIT	15	NB2	207	
NB4	THIRD ORDER BINARY DIGIT	16	NB4	208	
NB8	HIGH ORDER BINARY DIGIT	17	NB8	209	
PWI	POWER INDICATOR	6	PWI	213	

(1) ALSO CALLED DSS (DISTANT STATION STATUS)

EB1001

Figure 5-22. Communication Line Interface - RS232C/RS366

Each CONNECTION PORT interfaces to the communication line through a Modem and ACU (if auto dial-out is used). This system can have the following states:

- Open, that is, in a state where bit serial BDLC frames can be sent and received simultaneously or where the connection is established and the CONNECTION PORT is now ready to send or receive alternately as directed by the STATION.

- Closed, that is, in a state where no BDLC frames can be passed between the CONNECTION PORTS.

- Pending, that is, in one or more intermediate states between open and closed and not able to send or receive BDLC frames.

A pending CPD can be compared to the condition which exists when a voice telephone call has been dialed and is still ringing. An open CPD is analogous to that voice telephone call after it has been answered but with no conversation in process. A closed CPD is similar to the condition that exists after the phones have been hung up.

STATION DIALOG

The STATION DIALOG system is composed of a pair of STATIONS and a single CONNECTION PORT DIALOG. A STATION DIALOG can be:

- Open, that is, in a state where it is able to send and receive ROUTER FRAMES.

- Closed, that is, in a state where it is not able to send or receive ROUTER FRAMES.

- Pending, that is, in one or more intermediate states between open and closed and not able to send or receive ROUTER FRAMES.

Commands to the Station

These commands allow the Operations Interface, by way of the Station Level Manager, to control the stations. They originate with Operational Commands such as ESTABLISH CALL, Configurational Commands such as ADD STATION, Inquiry Commands such as STATION, and Station Manual Commands such as OPEN STATION DIALOG.

OCPD - OPEN CONNECTION PORT DIALOG

This command causes the station to establish a Connection Port Dialog with the remote station. The functions performed depend upon the communication line type.

DEDICATED
: The CONNECTION PORT turns on the local DCE and waits for an indication from the remote DCE.

AUTOMATIC DIAL OUT
The CONNECTION PORT establishes a dial connection.

MANUAL DIAL OUT
This command makes the CONNECTION PORT ready for establishing a manual dial connection or accepting an incoming call.

Refer to the Operations Interface commands ESTABLISH CALL, AWAIT CALL, and OPEN CPD.

CCPD - CLOSE CONNECTION PORT DIALOG

This command can be executed only when the STATION DIALOG is closed. All control circuits to DCE are turned off and all functions managed by the CONNECTION PORT are reset.

Refer to the Operations Interface commands CLEAR CALL and CLOSE CPD.

OSD - OPEN STATION DIALOG

This command can be executed only when the Connection Port Dialog is open. It initializes the BDLC protocol. The following functions are performed depending on the communication line type:

Dedicated -
The station sends a Set Asynchronous Balanced Mode (SABM) command (Refer to Section 9 - FRAME FORMATS) and waits for an Unnumbered Acknowledgment (UA) response, indicating that both the local and remote stations have gone into Asynchronous Balanced Mode. This is an operational BDLC mode in which both stations are ready for the transmission and reception of messages. If no UA is received, SABM is retried up to a limit, then the station is placed into a state in which it waits for a SABM from the remote station.

Automatic/Manual Dial Out -
Same as Dedicated above but does if no UA is received, SABM is retried up to a limit, and then the station is closed.

Refer to the Operations Interface commands ESTABLISH CALL, AWAIT CALL, and OPEN SD.

CSD - CLOSE STATION DIALOG

This command closes the Station Dialog in accordance with the BDLC protocol by sending a Disconnect (DISC) command. Refer to the Operations Interface commands CLEAR CALL and CLOSE SD.

ST - SEND TEST

This command is used by the host operator to diagnose the integrity of a link between this local host and a remote host. The SEND TEST command can be associated with an information field (TEXT) to be transmitted by the local station and returned from the remote station. Refer to the Operations Interface command SENDTEST.

The CONNECTION PORT must be open for the station to execute this command. The STATION DIALOG can be either open or closed.

The link level SEND TEST command allows diagnostic activity to take place

- Concurrent with normal network operation
- Independent of full or improperly attached queues in other parts of Network Services or Host Services
- Independent of potential errors in a Router routing table.

On full duplex circuits employing modems with "loop-back" features, the SEND TEST command allows maintenance personnel to distinguish the following kinds of possible failures at the link level:

- Data communication hardware/cables at the local host
- Modem at local host
- Telephone line to remote host
- Modem at remote host.

On half duplex circuits, the SEND TEST command can diagnose the link level for proper or improper operation but cannot isolate failures to the level of detail possible on FDX circuits.

SF - SEND FRAME

This command is used by the SLM to send a SLM control frame as an I frame to a remote SLM. The STATION DIALOG must be open to execute this command. Refer to the Operations Interface command VALIDATE AND ATTACH.

SET ATTRIBUTES

The SET ATTRIBUTES command allows the SLM to change the values of the STATION GROUP attributes. Refer to the Operations Interface commands ADD STATION, and MODIFY STATION.

GET ATTRIBUTES

The GET ATTRIBUTES command allows the SLM to obtain the current value of STATION GROUP attributes. Refer to the Operations Interface command STATION.

F-Response Timer Determination

In BNA, BDLC switched stations operate in a half-duplex mode. In order to resolve contention for the line, the two stations use different timer values for retrying. This is the F-Response Timer. When calls are made on switched stations, the initial selection of the values is based on which station placed the call, that is, on the incoming station having one value and the outgoing station having a different value.

After Greeting 1 messages are exchanged during the station level validation process the station whose node address is the higher of the addresses (Local Node Address and ONA in the received Greeting 1) uses the incoming timer value. The stations whose node address is the lower of the two uses the outgoing timer value.

This means that timer values are changed after receiving the Greeting 1 message approximately half of the time. However, it also means that there is no special processing involved in recovering pre-existing dialed connections during initialization, nor during manual dialing and answering, with regard to properly setting the F-Response timers.

Manual Dialing

Manual dialing and manual answering of calls are provided in BNA. The Station List entry in the SLM must be BDLC SWITCHED without ACU and the COMMUNICATION LINE TYPE in the BDLC Station Group must be set to MANUAL DIAL OUT.

Incoming calls, if they occur, are automatically answered by the BDLC Station when the COMMUNICATION LINE TYPE is set to MANUAL DIAL OUT and the station determines that Ring Indicator has turned on. If a call is manually answered, the protection of the Data Terminal Ready (DTR) to Data Set Ready (DSR) timer is lost. Refer to Figure 5-22 - Communication Line Interface - RS232C/RS366

When manually establishing a connection over BDLC SWITCHED lines, an Operator at one node must enter an ESTABLISH CALL command at the Operations Interface and call the remote station.

If the called station is already in an Await Call state, the calling Operator hears the answer tone. This indicates that the called station has automatically answered his call. The calling Operator then depresses the DATA mode button and the operation is complete. This is the (normal) case of a Manual-dial station calling an Auto-answer station.

If the called station is not in an Await Call state, the Operator at the called station answers (voice) the phone. If the two Operators agree (voice) to establish a connection, the called party enters an AWAIT CALL command at his Operations interface. After the called party has received a response to the

Await Call command, the called party communicates this fact (voice) to the calling party. Both parties enter the DATA mode by depressing the DATA mode button. The manual operation is now complete. This is the case of a Manual-dial station calling a Manual-answer station.

If the Auto-dial capability is used to call manual answer stations, establishment of a connection cannot be guaranteed. In general, auto-dial equipment terminates a connection if the called party does not provide the answer tone immediately after the phone is answered by the called party. This is one reason why switched stations not in use should always be left in an Await call state. However, if an Operator answers a telephone call and no Operator (voice) is at the remote end of the line, the Operator can attempt to answer the call from the remote node (computer). This is accomplished by entering an AWAIT CALL command at the Operations interface and depressing the DATA mode button as soon as possible.

The neighbor node validation and authentication features apply to both the manual dial and manual answer functions.

Automatic Call Unit (ACU) Capability

There are two STATION TYPES defined for BDLC SWITCHED Stations: with ACU (Automatic Call Unit) capability and without ACU capability. The SLM never selects a station without ACU capability that attempts an outgoing auto-dialed call. This is ensured by preventing the setting of an OUTGOING ENSEMBLE IDENTIFIER for such a station and rejecting an ADD CONNECTION - (Station or Ensemble) command (for the outgoing direction) that does not contain CALL DATA.

Likewise, the SLM never selects a station with ACU capability to make an outgoing manual dial call. This is ensured by checking the SLM Station Type attribute before opening the Connection Port on ESTABLISH CALL commands for BDLC SWITCHED stations which do not contain CALL DATA. If the Station Type is BDLC SWITCHED with ACU, outgoing connection type entries are referenced or built in the Neighbor Table. If the Station Type is BDLC SWITCHED without ACU, temporary outgoing connection entries are referenced or built in the Neighbor Table and the SLM waits for the call to be manually dialed.

To perform manual dialing on a BDLC SWITCHED Station with ACU, the station must be deleted and reconfigured as a BDLC SWITCHED Station without ACU before the manually dialed operation is performed.

X.25 STATION GROUP

Node Interconnection Using an X.25 PDN

A Public Data Network (PDN) can be used for the interconnection of BNA nodes. Interconnection via a PDN will include point-to-point circuits to provide node access to the PDN. Each attached node must conform to the communications interface defined in CCITT Recommendation X.25. X.25 defines the following 3 levels of interface:

- Physical level -- the physical, electrical, functional, and procedural characteristics to establish, maintain and disconnect the physical link between the DTE and the DCE;

- Link level -- the link access procedure for data interchange across the link between the DTE and the DCE;

- Packet level -- the packet format and control procedures for the exchange of packets containing control information and user data between the DTE and the DCE.

The X.25 interface defines a means of multiplexing multiple, independent traffic-streams over one physical circuit between node and VAN. In principle, the VAN is able to fully interconnect a network of up to 4095 nodes with each node supporting only one physical circuit. These nodes are interconnected not by physical circuits but by "virtual circuits."

Permanent virtual circuits (PVC) are similar to common dedicated or leased telephone circuits insofar as a connection is always present between the two end points (i.e., the two nodes). Virtual calls (VC) are similar to common dialed telephone circuits insofar as a connection between two end points is not always present and must be established and ultimately cleared according to some defined procedure.

X.25 Station Group Structure

The X.25 Station Group consists of a BDLC Station Group with another layer of function built on top of it. This other layer maps the X.25 logical channels to individual stations for use by the Router.

PHYSICAL AND LINK LEVELS

The X.25 Station Group supports an interface to an X.25 PDN. It relies upon a BDLC Station Group to provide physical and link levels. The BDLC Connection Group supports the link level while the rest of the BDLC Station Group supports the physical level. Communicating nodes employ separate physical links to the PDN. These links need not operate at the same baud rate. Hence the execution of the physical and link level protocols at one host is completely independent of the execution at the other node.

PACKET LEVEL

Logical Channels

Up to 4095 logical channels may be superimposed on a single physical circuit connecting DCE and DTE. Logical channels do not generally exist on an end-to-end basis through a PDN but rather exist strictly between DTE (the X.25 Station Group and DCE. The PDN employs proprietary DCE-to-DCE mechanisms to concatenate logical channels of communicating hosts to provide the apparent end-to-end path.

The concatenated logical channels will likely have different LCN's. This is because

- LCN's are selected by the DTE for outbound calls. Selection procedes downward from the highest numbered LCN in the list of LCN's assigned for outgoing calls (assignment is made at the time of subscription to the PDN).

- LCN's are selected by the DCE for inbound calls. Selection typically procedes upward toward the highest numbered LCN in the list of LCN's for incoming calls.

Furthermore, the sequence of operations and the assignment of packet sequence numbers to data packets may not necessarily be the same on each logical channel (depending on whether or not the PDN provides end-to-end significance).

Packet Level Protocol

The packet level protocol consists of:

- a set of packet formats defined to provide control functions such as call establishment, clearing, reset, and acknowledgement; supervisory functions to provide flow control; and data transfer.

- a set of rules governing the processing of received and transmitted packets.

Each of these packets is contained in the LINK INFORMATION UNIT of a BDLC frame as shown in Section 9, Frame Formats.

The physical level relies entirely on the link level for packet integrity. Hence, there is generally no retransmission capability in the packet level protocol. The only form of error recovery at the packet level is to reset or clear the call on a virtual circuit (reset the call on a permanent virtual circuit) and seek to re-establish another call.

LCN Service Algorithm

On input, the X.25 Station Group services LCN's on a first come - first served basis. On output, the Station Group allows LCN service to be predefined based upon the number of positions an LCN may be given in a table which is sequentially scanned to allocate service.

X.25 Optional User Facilities

Most of the optional user facilities be offered by PDN are unnecessary for a BNA user. Their functions can be performed within BNA. These include the following:

One Way Logical Channel Outgoing
Restricts the logical channel use to originating outgoing virtual calls.

One Way Logical Channel Incoming
Restricts the logical channel use to receiving incoming virtual calls.

Incoming Calls Barred
Incoming virtual calls are rejected by the DTE. The DTE can originate outgoing virtual calls.

Outgoing Calls Barred
The DTE does not send outgoing virtual calls to the DCE. The DTE can receive incoming virtual calls.

Closed User Group with Outgoing Access
Enables the DTE to belong to one or more closed user groups and to originate virtual calls to DTE's having the incoming access capability.

Closed User Group with Incoming Access
Enables the DTE to belong to one or more closed user groups and to receive incoming calls from DTE's having the outgoing access capability.

Incoming Calls Barred Within a Closed User Group
Permits the DTE to originate virtual calls to DTE's in this closed user group and precludes the reception of incoming calls from other DTE's in this closed user group.

Outgoing Calls Barred Within a Closed User Group
Permits the DTE to receive virtual calls from other DTE's in this closed user group and prevents the DTE from originating virtual calls to other DTE's in this closed user group.

Maximum Packet Length
The BNA X.25 Station operates with a packet size of 128.

Packet Sequence Numbering
The BNA X.25 Station operates with modulo 8 packet sequence numbering.

A BNA user of a PDN can subscribe to a Closed User Group Facility or a Bilateral Closed User Group Facility. These are described in the following paragraphs:

CLOSED USER GROUP FACILITY

A Closed User Group (CUG) is a set of X.25 Station Groups (DTEs) which can communicate with each other. This facility applies only for virtual calls. The facility is defined through an ADD CONNECTION or ESTABLISH CALL with CALL DATA command that contains specially formatted CALL DATA.

An X.25 Station Group can belong to more than one Closed User Group. The CUG used for a virtual call is indicated in the optional user facility parameters of either the CALL REQUEST or INCOMING CALL packets.

Note that if a Station Group (DTE) belongs to only one CUG, the indication of a CUG facility is not present in a CALL REQUEST or INCOMING CALL packet.

BILATERAL CLOSED USER GROUP FACILITY

A Bilateral Closed User Group is two X.25 Station Groups (DTEs) which can communicate with each other. This facility applies only for virtual calls (all SVCs). The facility is defined through an ADD CONNECTION or ESTABLISH CALL with CALL-DATA command that contains specially formatted CALL DATA.

An X.25 Station Group (DTE) can belong to more than one Bilateral Closed User Group. The Bilateral Closed User Group selected for a virtual call is indicated in the optional user facility parameters of either the CALL REQUEST or INCOMING CALL packet. The destination DTE address length field of the CALL REQUEST packet must contain all zeros.

NETWORK SERVICES MANAGER

GENERAL

The Network Services Manager (NSM) performs management functions which are global to the three functional levels of Network Services, the Port Level, the Router, and the Station Level. These functions include:

- Overall Network Services Host Initialization.
- Overall Network Services Node Initialization and shutdown.
- Interlevel control communications between the three major functional levels.
- Network Services Logging and Monitoring.
- The Network Services user process control interface.
- The Network Services Operations Interface.

FUNCTIONAL BLOCKS

Figure 5-23 shows the functional block diagram of the Network Services Manager. This is discussed in the following paragraphs.

Switcher Function

All control messages (Commands, Responses, and Reports) within Network Services are routed through the Network Services Manager. The Switcher function receives these messages then forwards them to their destination. Note that only control messages take this path, data messages pass directly between the user, the Port Level, the Router, and the Station Level.

The Switcher invokes the logging function and the Operations Interface Handler as needed for each message.

Attribute Handler

The Attribute Handler provides access to the NSM attributes. Upon command, this function returns the values of these attributes and stores new values (if valid).

User Interface

The User interface opens and closes user Subports, interrogates and sets port attributes, and reports errors back to the user.

Commands pass from the user through the User Interface Handler to the Switcher function, and the Responses pass through the User Interface Handler from the Switcher function back to the user. The User Interface Handler provides any format conversion or synchronization that is required.

Supervisor Functions

The Supervisor function coordinates the NSM phases and processes some of the reports received by the NSM. These reports are the Node Status Request Report from the Port Level; the Frame Report and the DNA Status Change from the Router; and the Attach, Detach, and Neighbor Status Reports from the Station Level.

Logging Function

The Logging function provides a method of recording specific events and accumulated counts for such purposes as billing, scheduling, reporting, or monitoring. This function is invoked for all messages received or generated by the NSM. The Logging function selects which of these messages is actually logged, based upon the actual message type and the current setting of the Logging Option.

The Logging function provides the capability for future retrieval of all items logged. Every message logged is stored in its entirety, thus all fields of every logged message are retrievable.

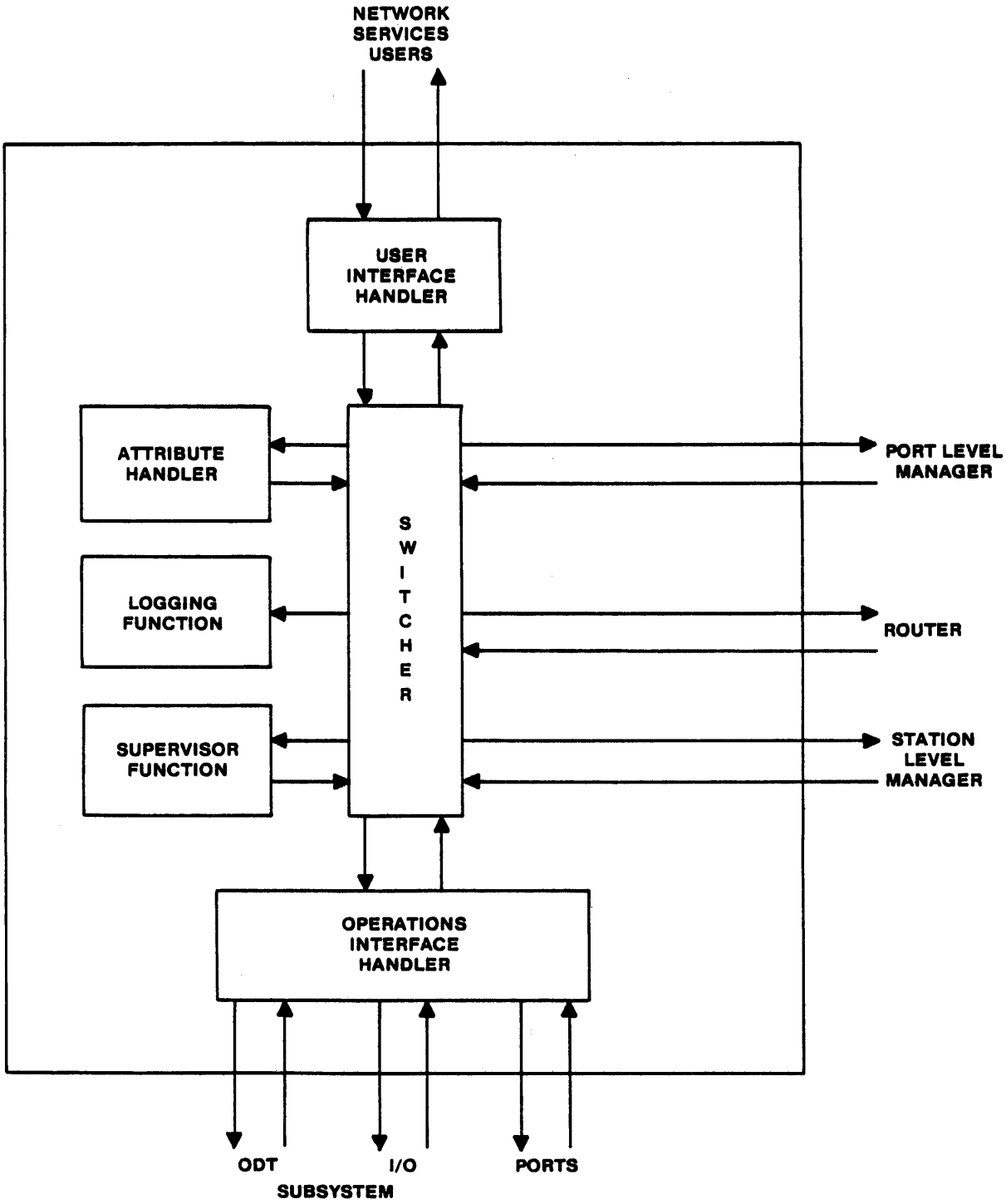


Figure 5-23. Network Services Manager Functional Block Diagram

The Logging function also receives the monitoring information generated by the functional levels. These results provide information about the operation of Network Services. This information is particularly useful for testing, maintenance, and performance monitoring.

OPERATIONS INTERFACE

General

The Operations Interface provides the capability for an agent to monitor and control a BNA node from an agent external to that Node. Messages, called Operations Interface Messages (or OIMs), may be used by the agent:

- to initialize the Node,
- to make inquiries about the status of the tables and attributes,
- to define and modify the node's network configuration,
- to change certain operating parameters,
- to initiate a specific action,
- to be informed of and to handle exception conditions,
- to shut down the Node, and
- to perform testing and maintenance.

The Operations Interface is administered by the Operations Interface Handler, a functional module which resides in the Network Services Manager.

Messages

There are three types of Operation Interface Messages - Commands, Responses, and Reports:

1. Commands are OIMs originated by the agent and sent to the BNA node. These messages inquire about the state of the node, change the operating behavior of the node by changing the values of attributes, define and modify the node's network configuration, and initiate some activity within the node.

2. Responses are OIMs which are generated by the node as a direct result of receiving a command, and these are sent to the agent. Every command generates a response.

3. Reports are unsolicited messages sent by the node to the agent, indicating the occurrence of some event in the node.

A detailed description of these messages can be found in the BNA Reference Manual, Volume 2 (Network Control). It provides a complete list of the messages, including syntax, semantics, response syntax, error messages, the operational phases in which they apply, and examples.

Agents

The agent can be a human operator working through an ODT, or it can be a file of messages for input, a printer (for output), or a user written program which sends and receives Operations Interface Messages. The agent can be local to the node being controlled or it can be remote. The BNA network itself is used to establish and maintain remote communication with remote agents.

ODT AGENTS

The ODT Agent is a human operator working through an ODT. There are four types of ODT agents:

- The operator at the local host's system ODT.
- The operator at the local host on an alternate ODT. An alternate ODT is an ODT which has been designated as an ODT agent with the Operations Interface TERM command. It can be a peer of the system ODT, or it can be the primary ODT agent.
- The operator at a remote host's system ODT.
- The operator at a remote host on an alternate ODT.

Responses to commands entered through an ODT are returned to that ODT. The ODT agent can specify that the responses be sent to a printer. Reports from the node are normally sent to the ODT agent.

As illustrated in Figure 5-24, the operators at the local host, either at the system console or an alternate ODT, communicate with the Operations Interface through their systems ODT Subsystem. Operators at a remote host communicate through that remote hosts ODT subsystem, to the ODT Subsystem in the local node through the BNA network using the Host Services ODT protocol, and finally to the Operations Interface Handler of the NSM in the node being controlled.

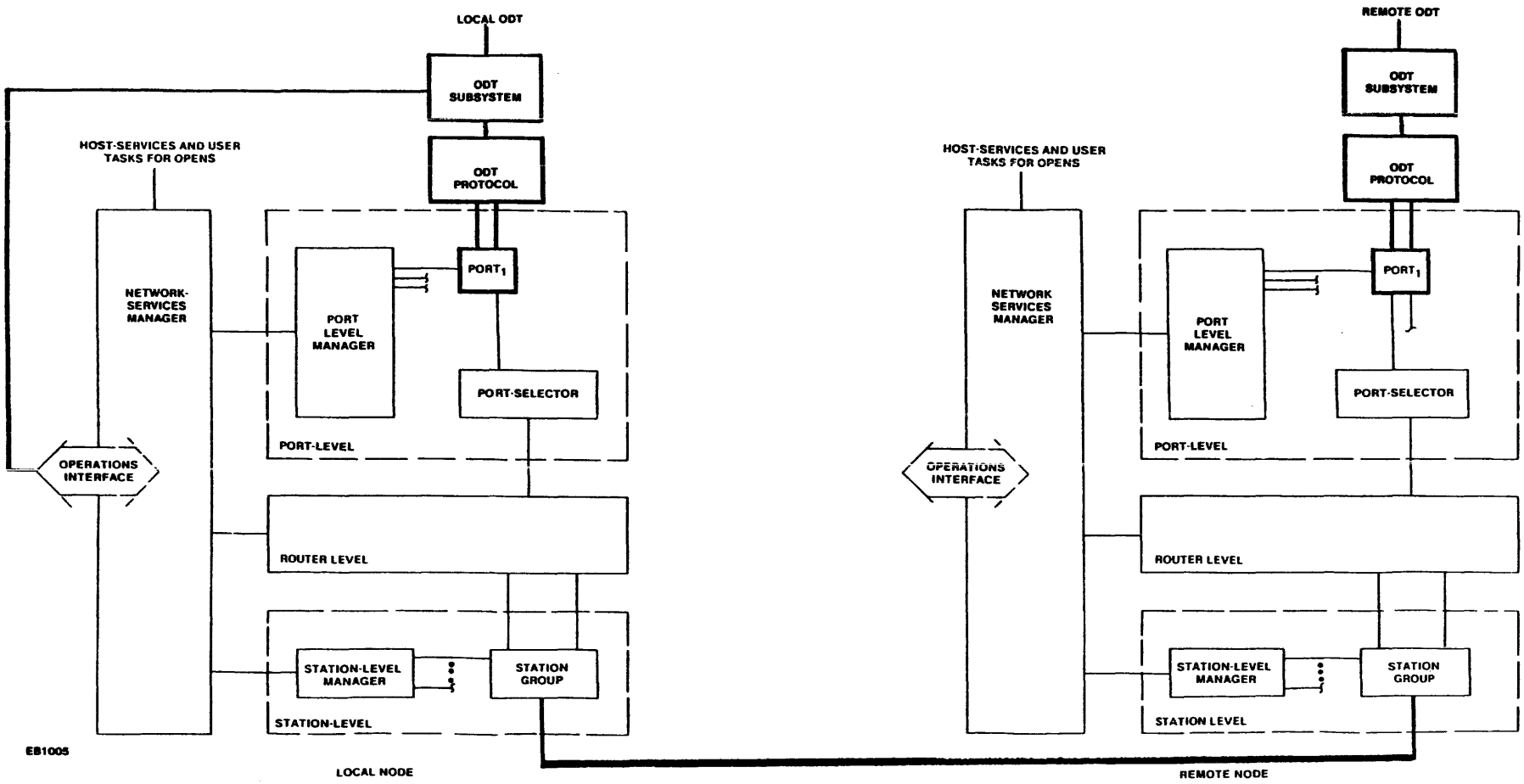


Figure 5-24. ODT Agents

OIM FILE AGENTS

OIM file agents are disk files that contain Operations Interface messages. Their operation is controlled by the Operations Interface SERVICE Command, for example, LOAD <file title>. The Initialization file is a special purpose OIM file agent used for initiating Network Services.

Responses to commands from file agents are sent to the ODT agent, except for those responses which are explicitly diverted or inhibited.

As illustrated in Figure 5-25, commands from file agents on the local host are passed to the Operations Interface Handler of the NSM via the local host's I/O subsystem. Commands from file agents on the disk files of a remote host are passed to the local host's I/O subsystem through the BNA network using the Host Services Logical I/O protocol, and then to the Operations Interface handler of the NSM in the node being controlled.

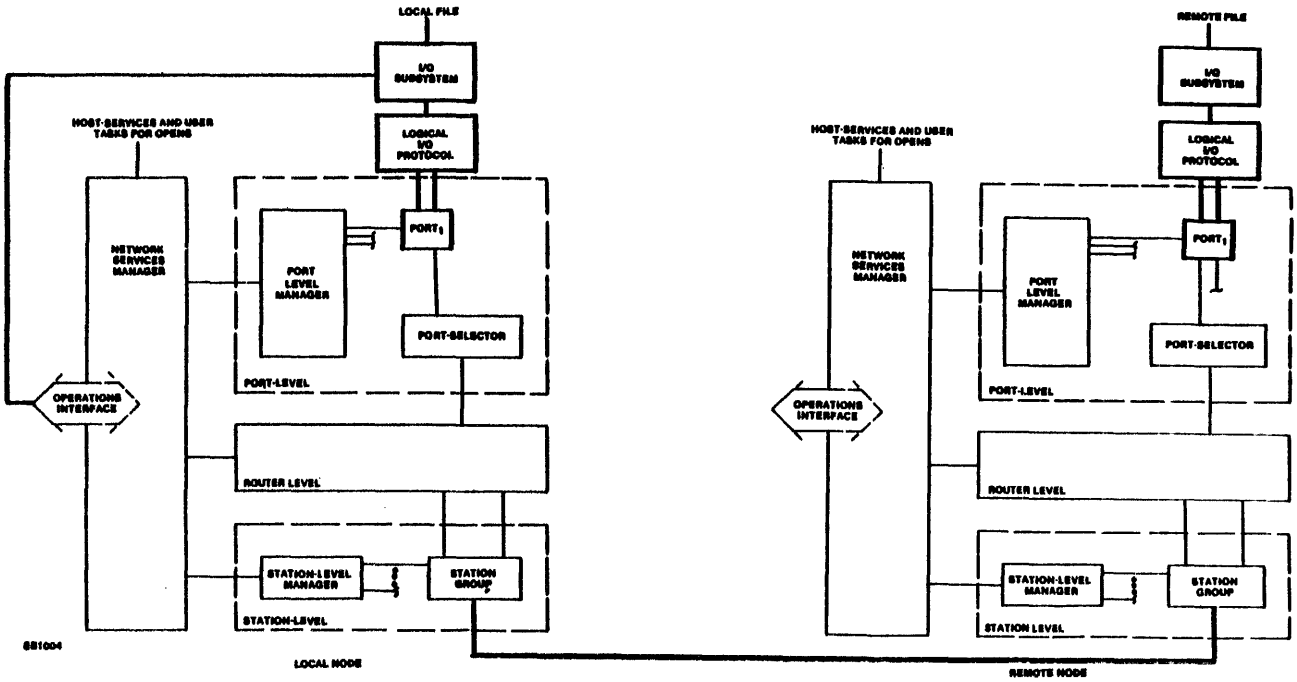


Figure 5-25. File Agents

PROGRAM AGENTS

These are user-written programs which can send and receive Operations Interface Messages. The agent can be local to the node being controlled or it can be remote.

Responses to commands from a program agent are returned to the program agent.

As illustrated in Figure 5-26, a program agent at either the local or remote host uses ports and Inter-Process Communication to communicate with the Operations Interface.

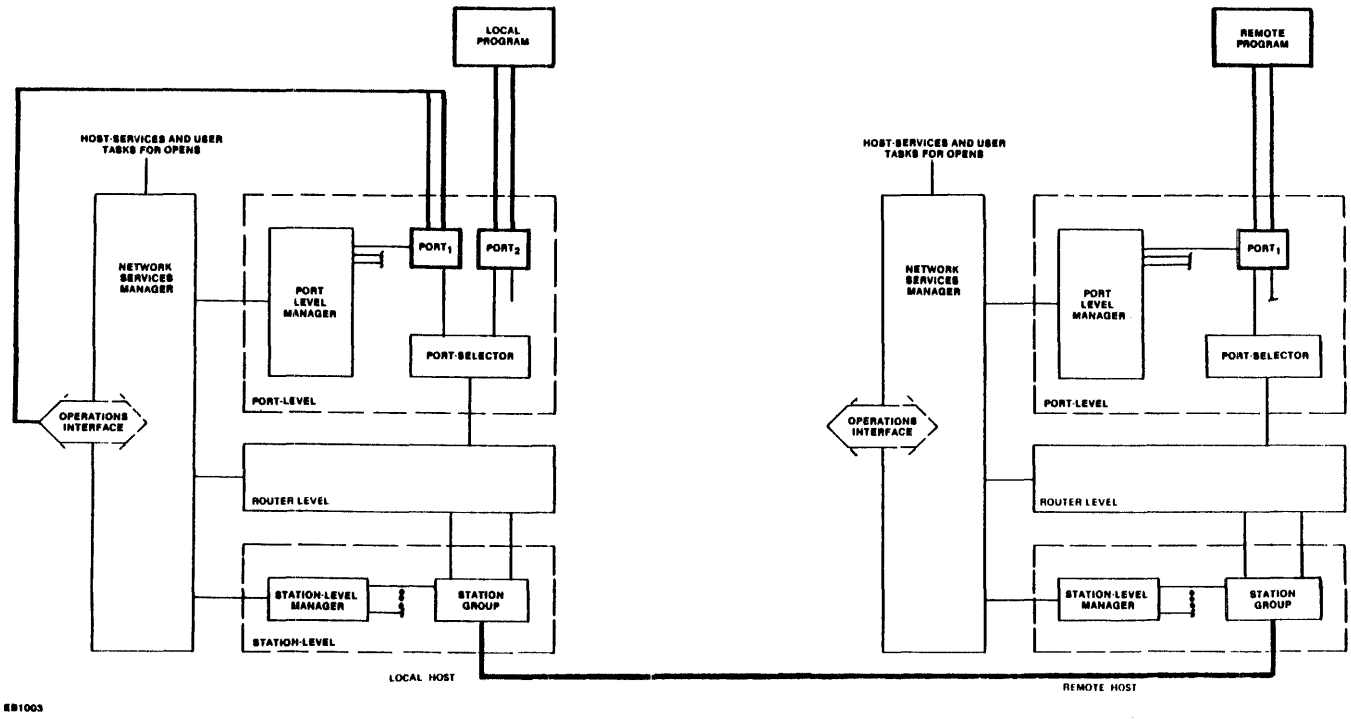


Figure 5-26. Program Agents

SECTION 6
NETWORK SERVICES INITIALIZATION

Network Services initialization is separated into two distinct parts: Host Initialization and Node Initialization. Host Initialization initializes the Network Services Manager (NSM) and the Port Level for local Inter-Process Communication (IPC) only. Node Initialization initializes the rest of Network Services (the Station Level and the Router) for remote network participation.

Host initialization occurs whenever the host system is initialized. During Host initialization any needed parameters are obtained in the same manner that the operating system software normally obtains its parameters. In addition, the NSM can also get a parameter that automatically starts node initialization.

Node initialization is started only upon entering the NET+ command. This command may be entered by the operator or it can automatically be entered during Host initialization. Node initialization is driven by a series of Operations Interface commands. These commands contain information about the local configuration and the network.

HOST INITIALIZATION

Host Initialization is coordinated by the NSM, and involves action only within the NSM and Port Level. Host Initialization starts when the host system is initialized. When completed, the NSM and Port Level both provide all the capabilities required to support local subport dialogs. They remain in this phase until a NET+ command is entered.

Any optional or required attributes are supplied to the NSM in a manner consistent with the methods normally used by the host system to perform similar functions. The NSM creates the Port Level, then sets both the NSM and the Port Level phases to ISOLATED. In this phase, most commands that would normally be handled by Network Services are rejected with a "NOT IN NETWORK MODE" error.

NODE INITIALIZATION

Because BNA defines a decentralized network in which there is no controlling node, the responsibility to initialize the network, both from the beginning and as each new node is added, lies with each participating node.

When a node enters a network, it is the responsibility of that node to initiate communication with each other node in the network. The node can be entering an already operating network, or initialization of a particular node can be part of bringing up an entire network.

Although it is possible to supply all of the required commands for node initialization directly via the ODT, it can be an unreasonable burden on the Network Services operator. For this reason, some semi-automatic assistance is

provided via the Initialization File. The Initialization File is a special input file. It is special in that its title is specified by the INIT FILE TITLE attribute and reading of the file is automatically initiated as part of node initialization. Refer to the Operations Interface command NET.

Initialization of a node is initiated by the Operations Interface command NET+. It is coordinated by the NSM and involves actions by the NSM, Station Level, Router, and Port Level. When completed, all of those levels are in normal operation, ready to handle Host Services or IPC functions with remote hosts. The operator is advised of entry into node initialization with the report "NODE IS NOW NETWORK INITIALIZING" and of the completion with the report "NODE IS NOW IN NETWORK MODE". The corresponding responses to NET inquiries are "NETWORK INITIALIZING" and "NETWORK OPERATING".

INIT FILE

The first action of Network Services node initialization is to prepare the Port Level for attribute entry and to start the other Network Services levels, the Router and the Station Level, in the attribute entry phase. The Port Level supports local subport dialogs throughout node initialization.

The NEXT INIT FILE TITLE is provided with the NET command. The NSM next takes the value in the NEXT INIT FILE TITLE and stores it into its INIT FILE TITLE attribute. If this value is non-null, the NSM locates and opens the file designated by this title as if a "LOAD <INIT FILE TITLE>" command had been entered from the Operations Interface. That is, if the file is not available, a "FILE NOT PRESENT" error message is displayed and the "LOAD" attempt is discontinued. If the file is made available at a later time, the operator can recover by entering a LOAD command specifying the same file name. The Initialization File "LOAD" operation is also aborted with an "IMPROPER FILE FORMAT" error if the located file is not formatted for an Operations Interface Message (OIM) input file. If the file is opened successfully, a "FILE LOADING IN PROGRESS" message is displayed and initialization commands are extracted from this source.

ATTRIBUTE AND CONFIGURATION ENTRY

The NSM converts each command to the corresponding internal commands and passes them to each of the Network Services levels. Each level then performs the actions requested by that command.

Because the default values for some of the Port Level attributes are dependent upon some of the Router attributes, the values of these attributes (MAX HOP COUNT and MAX RESISTANCE FACTOR) are transferred from the Router to the Port Level.

The NSM handles each command and distributes the internal commands that are generated until an END INIT command is detected by the Operations Interface.

If an END INIT command is not encountered, either because there was no command in the Initialization File or because no Initialization File was used, the operator is prompted for additional initialization input commands by "WAITING

FOR NETWORK INITIALIZATION INPUT" after every command, until the operator enters an END INIT.

STATION DIALOG INITIALIZATION

When the END INIT command is entered, the Station Level initiates Station Dialogs with its neighbors; opening permanent connections, recovering pre-existing dialed calls, and making outgoing calls. This call establishment and greetings sequence continues through Station Level Validation, ultimately resulting in ATTACH messages to the Router. When complete, the Station Level sends INIT COMPLETE control frames to its neighbors, puts its unused switched stations into an AWAIT CALL state, and notifies the NSM that it is in normal operation.

ROUTER INITIALIZATION

After attribute entry, the Router waits for the Station Level to initialize and send it ATTACH messages defining new station connections to neighbors, and for those neighbors to send it LINKCHANGE and NETCHANGE messages. The Router builds its table and sends out LINKCHANGE messages, but does not send NETCHANGE messages until an interval of time (Router attribute NODE UP TIMEOUT VALUE) has elapsed without receiving a LINKCHANGE or NETCHANGE message. By waiting this amount of time, all of the active neighbors should have sent to the local node all NETCHANGE messages they are going to send. Therefore, the NETCHANGE messages issued from this node should not be premature.

The Router reports all reachable nodes to the Port Level via the NSM, then reports to the NSM that it has gone to normal operation. On receipt of this report, the NSM goes to normal operation and node initialization is complete.

NODE SHUTDOWN

Node shutdown is the orderly process of a single node detaching itself from the network in which it has been operating. The shutting down of a BNA node can be defined only for the Node shutting down. The effect upon the rest of the network depends on the role of the node in the network. Nodes other than the one shutting down could become unreachable for some of the other nodes in the network.

Node shutdown is started only by a NET-, or NET- NOW, command. This command allows the option of either slow (NET-) or fast (NET- NOW) shutdown. It then proceeds as follows:

First, the Station Level is informed of the start of node shutdown and it then disallows any new connections to neighbors. ESTABLISH CALL commands are not honored and any received Greeting 1 messages are responded to with a negative Greeting 2 message - SHUTDOWN IN PROGRESS.

Next, if the shutdown is slow, no new non-local subport dialogs may be started, but existing non-local subport dialogs continue to completion. When all non-local subport dialogs have terminated, the Port Level informs the NSM that it has changed phase to ISOLATED.

Otherwise, if a fast shutdown is being performed, the Port Level deactivates all existing non-local subport dialogs and prohibits any new non-local subport dialogs. It then proceeds as in the slow shutdown case above.

When the NSM is advised by the Port Level of its change to ISOLATED phase, it tells the Router to shut down. The Router then sends NETCHANGE messages to all of its neighbors, indicating that this node cannot reach any node (resistance is MAXRF to all Nodes). The Router continues to process its input queues until activity subsides, then reports to the NSM. The Router then terminates processing.

Next, the Station Level closes all of its stations. It initiates a CLEAR CALL command for each BDLC and X.25 station. Each station's clearing process is the same as if a CLEAR CALL command had been received from the Operations Interface, except that any stations that would normally return to an AWAIT CALL state remain closed, and any SAVED stations stay SAVED. The Station Level then reports back to the NSM and terminates.

When the NSM receives this final Station Level report it changes its phase to ISOLATED. The NSM and the Port Level continue to support local subport dialogs.

Because slow node shutdown can continue for some time, during which it can be decided not to shutdown the node, a NET- CANCEL command is also defined. If this command is entered prior to the Router being informed of the shutdown the NSM instructs the Port Level to resume normal operation. If the Port Level accepts this command by returning a positive Response, the NSM instructs the Station Level to return to normal operation. Otherwise the CANCEL command is rejected with a "SHUTDOWN CANNOT BE CANCELLED" report.

NODE INITIALIZATION AS WITNESSED BY A NEIGHBOR

The above descriptions of node initialization have been concerned with the actions performed by the node that is initializing. However, the nodes in the network initialize themselves independently, in any order. Thus, a running, non-initializing node is part of the process of initializing one of its neighbors.

Normally, this node has tried to open stations to the neighbor and gotten no response, but the stations were left PENDING OPEN, waiting for the neighbor.

When the station receives a link-reset message (SABM in BDLC), it replies with its acknowledgment message (UA in BDLC) to the neighbor. The SLM then performs Station Level validation (using station greetings) and if it is successful, it then reports to the Router with an ATTACH via the NSM, and marks the neighbor as initializing.

When the Station Level receives the INIT COMPLETE control frame from the initializing neighbor, it marks the neighbor as operating.

After the Router receives the ATTACH, it sends and receives a LINKCHANGE message with the new neighbor. It then sends its NETCHANGE messages, and receives NETCHANGE messages. The Router reports the node reachable to the Port Level via the NSM.

Lastly, the Port Level, after receiving the reachability message, performs Port Level validation on the new Host(s).

NODE RE-INITIALIZATION AS WITNESSED BY A NEIGHBOR

Another case that occurs is for a neighbor node to re-initialize itself, without the local node detecting that it had stopped. As part of this re-initialization, the neighbor node loses all prior state information. That is, it has no knowledge of the fact that it was previously running, it only knows that it is currently initializing.

In this case, the local node status for the re-initializing neighbor node is incorrect because it was not yet known that the neighbor had gone down.

The stations are OPEN when the station receives an unexpected link-reset message (SABM in BDLC). Any unacknowledged output is prepared for retransmission and the Station Group reports STATION DIALOG REOPENED to the SLM before it sends the acknowledgment message (UA in BDLC).

When the SLM receives the STATION DIALOG REOPENED Report from the Station Group, it performs Station Level validation (using Station Greetings). In this case, the received greeting indicates that the neighbor is initializing, not resetting the link, so the SLM marks the neighbor as initializing. If this is the first re-initialization link-reset from that neighbor, a link-reset is done on all other stations to that neighbor, and a NEIGHBOR RESTART report is sent to the Router (via the NSM).

Because the neighbor is now marked as initializing, future stations that open from that neighbor (or fail to open) do not cause a link-reset.

When the Router receives the NEIGHBOR RESTART message, it marks the neighbor as restarting and starts a timer.

As Station Level validation for each link to this neighbor is complete, the SLM reports to the Router with an ATTACH (via the NSM). The Router then sends its LINKCHANGE and NETCHANGE messages to that neighbor, and receives a LINKCHANGE from it.

When the Station Level receives an INIT COMPLETE control frame, it marks the neighbor as operating.

When the Router receives a NETCHANGE from that neighbor, or its timer expires, the Router updates its tables, and reprocesses the output messages to that neighbor.

SECTION 7
ACCESS CONTROL

Access control in a BNA network is the means by which each node and host can control access to its resources from nodes, hosts, and users elsewhere in the network. Since hosts in a BNA network are peers, access control in a BNA network is a distributed function. Each node and host is responsible for controlling access to its own resources. Extensive options permit the Network Operations Manager at each node/host to choose the types and amounts of protection for that node/host.

There are two major types of access control:

Control of access by other nodes and hosts; and,

Control of access by individual users at those hosts.

Validation and authentication are used to control access by other nodes and hosts. Validation controls which nodes and hosts are allowed access. Authentication verifies that a node or host really is the node or host that it claims to be. Access control by individual users at that host is controlled with a Hostname/Usercode pair. The usercode is authenticated at the remote host when the user logs on to the remote host with the usercode/password by which the user is known at that host.

Some implications of this philosophy are:

1. In its list of valid users, each host maintains a hostname/usercode for those users whose requests for service may originate at other hosts in the network.

2. Limitations on resource access or usage set by one host do not affect access to resources at other hosts.

3. Each host may expand or revoke the privilege and capabilities of any user without necessarily notifying other hosts in the network.

The various validation functions permit the Operations function at a node to pre-define which hosts, nodes, and neighbors are acceptable hosts, nodes, and neighbors. Validation is optional and can be performed at all 3 levels of Network Services; Port, Router, and Station. Validation is applied when communications are initiated with a node/host.

Station Level validation permits Operations to describe which nodes are allowed access as neighbors, and which stations are to be used for those connections.

Validation at the Router level permits Operations to describe those nodes to which the Router will route frames.

Port Level validation permits Operations to describe which remote hosts are allowed access for port level dialogs.

In summary, the result of acceptance or rejection of a node by the validation function at each level has the following characteristics:

- If a Host is rejected by the Port Level, then no subports are opened with that Host for any reason, including Operations Interface ports. As a direct result of this, no user at that Host can gain access to any local resources, and no local Network Services user can communicate with that Host.

- If a node is rejected by the Router, then no traffic is routed to that node, regardless of whether the traffic originated locally or remotely (Note, however, that no check is made on traffic routed from that node to other valid nodes).

- If a node (neighbor) is rejected by the Station Level, then no Station Level dialog is opened with that node, i.e., it is not accepted as a neighbor.

The node and host authentication functions verify that a node really is the node it claims to be by comparing the received password with a pre-stored password for that node. Password checking can be performed at the Station Level between neighbor nodes, and at the Port Level during the establishment of PLM to PLM dialogs between hosts. Like validation, authentication is applied when communication is begun with the remote node/host. For each node/host which is to be authenticated, two passwords are maintained at the local node/host; the password that must be sent to the remote node/host in outgoing control frames, and the password that is expected in incoming control frames from the remote node/host. The passwords are included in the Station Level Greeting 1 control frames, and in the Port Level PLM Identification frame.

Individual users at a host are known elsewhere in the network by their hostname/usercode pairs. A hostname/usercode pair is unique in the network. Even if users on two different hosts have the same usercode, they have different hostname/usercode pairs because the hostnames are different. Users on the same host in the network have different hostname/usercode pairs because their usercodes are different. The hostname/usercode pair is used to check a remote user's access privileges using the same mechanism that is invoked for local usercodes. The rules for access at a host for a user logged on to that host apply to remote users as well.

An individual user at a remote host seeking access to a subport uses his hostname/usercode pair to establish a match to that subport. The hostname/usercode pair is also used in the initiation phase of Host Services protocols.

CONTROL OF ACCESS FROM OTHER NODES AND HOSTS

Validation and authentication are used to control access by other nodes and hosts. Validation controls which nodes and hosts are allowed access. Authentication verifies that a node or host really is the node or host that it claims to be.

VALIDATION

The following paragraphs describe validation at the Station, Router, and Host levels. Validation options are discussed, followed by some examples of validation in a network.

Station Level Neighbor Node Validation

There are two classes of neighbor node validation performed within the Station Level:

- Validation using Station Greetings after the Station Dialog is opened but before the station is attached to the Router.
- Validation performed during the establishment of a Station Dialog (e.g., validating parameters in an X.25 Incoming Call Packet).

VALIDATION USING STATION GREETINGS

The neighbor node validate function is performed by comparing the Origin Node Address (ONA) in received Greeting messages against Neighbor Node Addresses (NNA) in the Neighbor Table. The Operations Interface command ADD CONNECTION pre-stores the NNA's into the table. The Operations Interface command ESTABLISHCALL may also put an NNA in the table, but on a temporary basis, only for the duration of the call.

Validation of a connection occurs when a Greeting 1 message is received. When either ALL NEIGHBOR VALIDATE is TRUE or NEIGHBOR VALIDATE (for the station used to make the connection) is TRUE, the Origin Node Address is checked in the received Greeting 1 message.

If the Station's connection type is PERMANENT or INCOMING, the ONA must equal an NNA in the Neighbor Table. A permanent/incoming connection type entry referencing this station or a permanent/incoming ensemble to which this station belongs is required for the connection to be valid.

Validation is always performed on outgoing switched calls (both outgoing BDLC switched calls and outgoing X.25 virtual calls). This is what may be termed a "wrong number" check.

If validation fails, the call is considered invalid, a negative Greeting 2 message is sent in response to the received Greeting 1 message, and the line is disconnected if possible.

If the call has passed the validation criteria, the NNA is set in the Station, the Station-ID is added to the proper connection type entry in the Neighbor

Table and a positive Greeting 2 message is sent in response to the received Greeting 1 message.

Router Validation

Node address validation is the process by which the Router checks that a node address may be added to its tables. When ROUTER VALIDATE is FALSE, any Subject Node Address contained in an incoming NETCHANGE message is valid and is added to the router tables. When ROUTER VALIDATE is TRUE, the Operations Interface commands ADD HOST and ADD NODE define the nodes to be added to the router tables, and incoming NETCHANGE messages result in error conditions if their Subject Node Addresses are not already in the Routing Table Info (RTI).

Host Validation

The Host Validation function is performed by comparing a Hostname, Node Address pair received in a PLM-IDENTIFICATION Control Frame to those in the Remote Hosts table. The ADD HOST command puts the Hostname, Node Address pairs in the Remote Host table, pre-defining them as valid. The Node Address portion is not in the control frame, but is extracted from the Router header associated with the control frame.

Validation Options

Each of the three functional levels has an attribute that controls the actions of that level whenever it first communicates with a corresponding level at another node. These are called HOST VALIDATE (Port Level), ROUTER VALIDATE, and ALL NEIGHBOR VALIDATE (Station Level). If an option is true, a validation function is invoked when communication is first established at that level. As part of this validation procedure it is required that each new node be previously specified as valid. If not, communication is rejected at that level. Rejection at any one level does not affect validation at any other level. If VALIDATE is false, then that level accepts any new nodes/hosts by simply adding the new node/host to its tables.

In addition to the Network Services global VALIDATE options, the Station Level has attributes called NEIGHBOR VALIDATE that apply to individual stations. These are set as a byproduct of the ADD CONNECTION commands. The ADD CONNECTION (INCOMING/PERMANENT) command for an individual station sets NEIGHBOR VALIDATE to TRUE for the station. The ADD CONNECTION (INCOMING/PERMANENT) command for an ensemble sets NEIGHBOR VALIDATE to TRUE for all of the stations in the ensemble.

The Operations Interface command VALIDATE sets the VALIDATE attributes in the functional levels as follows:

command option			port level	router	station level
VALIDATE	=	NONE	F	F	F
VALIDATE	=	NEIGHBOR	F	F	T
VALIDATE	=	HOST	T	F	F
VALIDATE	=	HOST, NEIGHBOR	T	F	T
VALIDATE	=	ALL	T	T	T

(T = TRUE, F = FALSE)

It is not possible to set ROUTER VALIDATE to TRUE without also setting those of the Port and Station Levels to TRUE.

VALIDATE - ALL

For VALIDATE-ALL, all three levels have their VALIDATE attribute set true. In this mode of operation every node and host must be authorized to communicate at each level.

VALIDATE - HOSTS, NEIGHBORS

In this option, HOST VALIDATE and ALL NEIGHBOR VALIDATE are set true, but ROUTER VALIDATE is set false.

In this mode of operation, Port Level communications and neighbor connections are still restricted to their own sets of authorized hosts and neighbors, but the Router accepts any compatible node. Thus, transit traffic is permitted from any neighbor and forwarded, if possible.

By including or excluding a specific host from the lists of authorized hosts and neighbors, it is possible to independently accomplish the following actions:

- specifically include or exclude a host from access to the local Port Level,

- specifically include or exclude a node as a direct neighbor.

VALIDATE - HOSTS

In this option, ROUTER VALIDATE and ALL NEIGHBOR VALIDATE are false, HOST VALIDATE is true.

This mode of operation is selected if it is only desired to restrict the hosts capable of Port Level communications. Specifically, all transit traffic to known compatible nodes is allowed, and any compatible node is allowed to be connected as a neighbor.

VALIDATE - NEIGHBORS

In this option, only ALL NEIGHBOR VALIDATE is set true.

Using this mode of operation, it is only possible to restrict the nodes that are allowed to be direct neighbors. As in the previous two cases, all Router traffic is allowed, but in addition all hosts are allowed to establish Port Level communications.

VALIDATE - OFF

In this option, all three levels have VALIDATE false.

In this mode of operation, any compatible node that can access the local node, in any manner, is allowed to do so both for Port Level communications and for transit traffic. This is the default value of the VALIDATE option.

Validation Examples

The following initialization files are examples of the types of control on the network traffic paths that can be exercised by Host A (Node 1) using the validation functions. Refer to Figure 7-1 and the BNA Reference Manual, Volume 2 (Network Control).

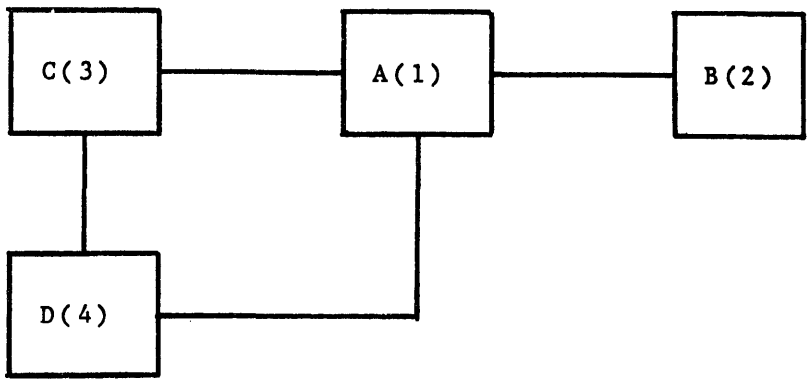


Figure 7-1. Sample Network

Assume that Hosts B, C, and D have VALIDATE=NONE and that all attempt to establish full communications on all levels with all of the hosts and nodes.

1. Node A Initialization file example to allow local port communications with all hosts, routing to all nodes, and allow all connected nodes as neighbors.

```
LOCALIDENTITY = #1;
VALIDATE = ALL;

ADD HOST B #2;
ADD HOST C #3;
ADD HOST D #4;

ADD STATION ATOB TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOC TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOD TYPE=BDLCDEDICATED (SPEED=9600, etc);

ADD CONNECTION PERMANENT WITH B BY ATOB;
ADD CONNECTION PERMANENT WITH C BY ATOC;
ADD CONNECTION PERMANENT WITH D BY ATOD;

ENDINIT;
```

2. Node A Initialization file example to disallow all communications with Host B (Node 2).

```
LOCALIDENTITY = #1;
VALIDATE = ALL;

ADD HOST C #3;
ADD HOST D #4;

ADD STATION ATOB TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOC TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOD TYPE=BDLCDEDICATED (SPEED=9600, etc);

ADD CONNECTION PERMANENT WITH C BY ATOC;
ADD CONNECTION PERMANENT WITH D BY ATOD;

ENDINIT;
```

3. Node A Initialization file example to allow transit traffic only for a node (4), but disallow it as a direct neighbor, and disallow port dialogs with its host (D).

```
LOCALIDENTITY = #1;
VALIDATE = ALL;

ADD HOST B #2;
ADD HOST C #3;

ADD NODE   #4;

ADD STATION ATOB TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOC TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOD TYPE=BDLCDEDICATED (SPEED=9600, etc);

ADD CONNECTION PERMANENT WITH B BY ATOB;
ADD CONNECTION PERMANENT WITH C BY ATOC;

ENDINIT;
```

4. Node A Initialization file example to allow transit traffic only for a node (4), and to allow it as a direct neighbor, but to disallow port dialogs with its host (D).

```
LOCALIDENTITY = #1;
VALIDATE = ALL;

ADD HOST B #2;
ADD HOST C #3;

ADD NODE   #4;

ADD STATION ATOB TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOC TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOD TYPE=BDLCDEDICATED (SPEED=9600, etc);

ADD CONNECTION PERMANENT WITH B BY ATOB;
ADD CONNECTION PERMANENT WITH C BY ATOC;
ADD CONNECTION PERMANENT WITH D BY ATOD;

ENDINIT;
```

5. Node A Initialization file example to allow transit traffic to a node (4) and local port communications with its host (D), but not to allow it as a neighbor.

```
LOCALIDENTITY = #1;
VALIDATE = ALL;

ADD HOST B #2;
ADD HOST C #3;
ADD HOST D #4;

ADD STATION ATOB TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOC TYPE=BDLCDEDICATED (SPEED=9600, etc);
ADD STATION ATOD TYPE=BDLCDEDICATED (SPEED=9600, etc);

ADD CONNECTION PERMANENT WITH B BY ATOB;
ADD CONNECTION PERMANENT WITH C BY ATOC;

ENDINIT;
```

AUTHENTICATION

Station Level Authentication

Neighbor node authentication is performed by comparing the password in a received Greeting 1 message with a prestored password for the node identified by the Origin Node Address in Greeting 0 and 1 messages. The password checking feature is a user option. If Operations pre-stores an incoming password for a given other node, that password must be included in any Greeting message from that neighbor. If Operations does not store an incoming password for a given other node, any password or null is accepted in incoming Greeting messages from that node. This is the default. In outgoing Greeting messages the SLM includes a password for the outgoing direction if one has been pre-stored for that node. Otherwise, a null password (the default) is sent.

The SLM at each node uses two passwords for each neighbor node. One password is sent to the neighbor node and the second password is used to check the password received from the neighbor.

The node authentication function is most useful when both neighbor node validation and password checking are required by and for all links and nodes within the network. Another useful combination is to provide neighbor node validation and password checking functions on certain links (for example, high performance links) and not on others.

Port Level Authentication

The Port Level Authentication function is performed by comparing the password received in a PLM IDENTIFICATION control frame to a pre-stored value for that host. The frame is authenticated if the password either exactly matches the pre-stored REMOTE PASSWORD TO ME or if the REMOTE PASSWORD TO ME specified for that remote host is null. MY PASSWORD TO REMOTE is sent to the remote host in PLM IDENTIFICATION control frames.

Each node/host can have two passwords for each other node/host. The SLM and PLM both use the same set of passwords for a given node/host.

STATION LEVEL GREETINGS

To provide for both the validation and authentication functions, the SLM must exchange Greeting messages without revealing any useful information about the local node until it has performed validation and authentication on the remote node attempting connection. This requires a hierarchy by which to exchange the sets of Greeting messages. The protocol for SLM Greeting message exchange is described in the following paragraphs.

The SLMs exchange Greeting 0 messages as soon as the station dialog is open. The Greeting 0 message tentatively identifies the potential neighbors to one another by node address. It provides for those cases when a node does not know the neighbor address, as when an await call command is entered or when an establish call command is entered without a neighbor parameter.

Once Greeting 0 messages are exchanged, both SLMs know the node address of the other node. Thus which password to use is established; then the Greeting 1 - Greeting 2 greeting functions can begin.

On switched lines, one party must call the other. Greeting 0 messages are exchanged. The calling party (processing an establish call command) will send a Greeting 1 message (containing the calling party's node address and the appropriate password). The called party (processing an await call command) checks the validity of the entire Greeting 1 message (including password) before sending a Greeting 1 message in return. In this manner, the password mechanism does not reveal any information until it has verified the calling neighbor. Since the hierarchy for password exchange on switched lines is based on the entry of an establish call command, the process of manually dialing a phone call must be preceded by an establish call command.

For dedicated lines, there is no Calling/Called relationship at the SLM. Both SLMs exchange Greeting 0 messages. Then they exchange Greeting 1 messages. Greeting 2 messages are sent after the validation and authentication functions are performed. Greeting 1 message exchanges on permanent connections do not require checking the remote node's validity prior to Greeting 1 message interchange.

CONTROL OF ACCESS FROM INDIVIDUAL USERS AT A REMOTE HOST

Once a remote Host has been granted access to the local Host, additional access controls on individual users at that remote Host come into play. The basic mechanism is the Hostname-Usercode pair. A user signs onto a Host in the usual manner using Usercode and Password. If the Host grants the user access to its resources, that user may now be known throughout the network as that Usercode at that Host (and that is the Hostname-Usercode pair). Other Hosts in the network maintain lists of Hostname-Usercode pairs in their user data files (or equivalent), and these lists are used to control access to their resources. Once a Host has found that an incoming Hostname-Usercode pair is authorized (that is, is in its list), the mechanism to control access to local resources is identical to the control of access by local users. Some details vary among the various implementations of BNA. In some systems, after the incoming Hostname-Usercode pair is authenticated, the incoming Usercode is used as if it were the local Usercode. Note that this requires that Operations management coordinate the use of Usercodes which are to be used across the network, but also note that no coordination is needed for Usercodes to be used only locally. In large systems, an additional "alias" mechanism is provided. After a Hostname-Usercode pair has been authenticated (found in the local tables), a local alias for that remote Usercode can be assigned. The same Usercode could be assigned, and this acts exactly the same as the mechanism described above, but an entirely different local Usercode could be assigned, thus reducing the amount of coordination needed.

HOSTNAME/USERCODE CONTROL IN PORT MATCHING

A subport opened by a user process at a remote host seeking access to a local subport must establish a "match" to that subport. Refer to Section 5, Port Level for details. The subports match if they are "complementary" and if the user process has "access" to the subport. A subport dialog between the two user process' subports is established when the matching is completed.

Conditions FOR Complementary Subports

Two subports are complementary if all of the following conditions are true:

- PORTNAME A = PORTNAME B
- YOURNAME A = NULL OR YOURNAME A = MYNAME B
- YOURNAME B = NULL OR YOURNAME B = MYNAME A
- YOURHOST A = NULL OR YOURHOST A = MYHOST B
- YOURHOST B = NULL OR YOURHOST B = MYHOST A

Accessibility

Each host performs security checking for its own subports as follows:

- If the value of the SECURITYTYPE attribute in the subport's port is PUBLIC, the subport is accessible to any user process.

- If the value is PRIVATE, the value of its YOUR USERCODE must match the usercode of the process offering a complementary subfile for matching. The value of its YOURNAME forms the other half of the hostname/usercode pair.

- If the value is GUARDED, the result of the security check depends on the contents of the file named by the SECURITYGUARD attribute. Refer to the Large Systems I/O Subsystem Reference Manual.

HOSTNAME/USERCODE CONTROL IN HOST SERVICES PROTOCOLS

Generally, usercodes are passed from the initiating host to the cooperating host in the initiation phase of the protocol. The hostname of the initiating host, along with the usercode, forms a hostname/usercode pair which is used by the cooperating host to decide whether to accept the request for service. A usercode/password, authenticated at the receiving host, can be transmitted in some of the protocols as an alternate means to identify the user.

Operator Display Terminal (ODT) Protocol

This function allows operator input messages to be transferred to a remote host and responses to be returned. A usercode is sent to the remote host with each message. For requests made via the ODT, if no usercode is available (i.e., terminal usercode), the host usercode is used. The remote host accepts the usercode as pre-authenticated by the originating host, and tests it for its privilege (i.e., whether it is a "systemuser" or not), that the hostname/usercode pair is in its tables, etc.

In addition, a usercode/password can be included as part of the operator inquiry. This, like the rest of the operator input, is not checked at the sending host. It is a usercode/password to be authenticated at the receiving host.

Remote Tasking Protocol

Tasks may be initiated at a remote host with this Host Services function. The protocol requires that the parent job or task be running under a usercode. This usercode and the host at which the parent task is running make up the hostname/usercode pair used by the receiving host.

Job Transfer Protocol

This function allows a user to transfer an entire job deck to a remote host for interpretation and execution. If there is a usercode associated with the job performing the Job Transfer, that usercode is sent to the remote host, and it and the host from which the transfer originated make up the operable hostname/usercode pair. The job deck itself can contain a usercode/password to be authenticated by the receiving host.

Logical I/O Protocol

This function supports logical I/O across the network to allow a user to read and write files at another host. The protocol requires that a usercode be associated with the task initiating the logical I/O operations, and this usercode becomes part of the operable hostname/usercode pair. If the value of SECURITYTYPE for the file is PRIVATE, access is granted only to a specified usercode. If it is GUARDED, access is granted only if the requestor satisfies the requirements of the guardfile.

In addition, a usercode can be part of the file title. It is a usercode at the host where the file resides and is used for file identification only.

Station Transfer Protocol

This function is used to logically connect a terminal on one host with a process running on another host. In the CONNECT case, where the initiating host has a terminal to connect to a remote process, the usercode of the user at the terminal is passed to the remote host. In the ATTACH case, where the initiating host has a process to attach to a remote station, the usercode of the process requesting the attachment is sent.

In addition, after the terminal is transferred, the user at the terminal must log on to the process host as though the terminal were directly connected to that host.

Status Change Protocol

No usercodes are required for this protocol. It is initiated only to support other, already authorized, protocols.

USER ACCESS TO OPERATIONS INTERFACE COMMANDS

An agent (usercode) at the local or a remote host can be authorized (or denied the authority) to enter classes of Operations Interface commands, controlled by the command AUTHORIZE. The agent which grants or removes a specified authorization must possess the authority itself. The classes are ATTRIBUTEENTRY, CONFIGURATION, OPERATION, MANUAL, INQUIRY, ALL, and NONE. Refer to the BNA Reference Manual Volume 2 (Network Control) for a breakdown of the command classes.

PROGRAM AGENT CONTROL

Communications from a program agent, either at the local or a remote host, to the Operations Interface must be explicitly initiated by the Operations Interface PROGRAM command. These communications are further controlled by other Operations Interface commands, MAXPROGRAMAGENTS and PROGRAMAGENTSECURITY. The PROGRAMAGENTSECURITY command can require the program agent to match a specified usercode or to match a specified guardfile.

SECTION 8

LOGGING AND MONITORING

LOGGING FUNCTION

The BNA logging function provides a method of recording specific events for such purposes as billing, scheduling, reporting, performance monitoring, testing, and maintenance. BNA logging is in addition to any other system logging features.

The function of logging is performed by the NSM based upon information provided within its own level and from the other Network Services levels. The log entries are made to the log at the local node.

The Logging function selects which of the messages or reports are actually logged based upon the message type and the setting of the Logging Option. The Logging option may either be: OFF, MINIMUM, STANDARD, or MAXIMUM. These allow more or less logging as desired, and OFF is no logging at all. The options are invoked using the Operations Interface command LOGGING.

As mentioned above, the NSM performs all logging, however, each level is responsible for its own logging requirements. The information to be logged is passed to the NSM by means of reports and responses. The logging and monitoring functions performed by each level are discussed in the following paragraphs.

Due to differences in hardware and software implementations of BNA some differences in logging procedures may be observed. Refer to individual system BNA documentation for specific detail.

PORT LEVEL LOGGING

Logging information from the Port Level is intended for later use by the user. The log records include information on usage which may be logged periodically, or at the time that a resource is released. They also contain historical information of Port Level changes, including errors that are collected from Port Level responses and reports to the NSM, and commands to the Port Level from the NSM.

Where logging is done due the expiration of the periodic logging interval, all open subports are polled to obtain the data needed for logging. The logging period is determined by the setting of the PORTLOGINTERVAL attribute. This attribute can be set using the Operations Interface command PORTLOGINTERVAL.

The information logged for the subports under conditions of closure or periodic logging, include the following information:

- Using Process ID
- Port address
- Subport index
- Open Timestamp
- Close Timestamp
- Subport Attribute Value List, including:
  - Your Host Name
  - Your Name
  - Your Usercode
  - Logging info
    - messages sent and received
    - segments sent and received
    - number of retransmissions
    - number of control frames sent

Other Port Level information that can be used for logging state changes and errors are:

- Remote Node Address
- Remote Host Name
- Termination Reason
- Correct Max Segment Size
- Errors that are reported are:
  - Disconnected
  - Frame Format Errors
  - Subport Error
    - No room for last message
    - Data lost
    - No match
    - Unreachable host

ROUTER MONITOR AND LOGGING

For the various purposes of billing, network maintenance, analysis of network operation, etc., it is necessary to record Router occurrences in a log. The Router Monitor function, in conjunction with a monitor function in the Network Services Manager (NSM), performs the required logging.

Tables and Other Attributes

At Network Services initialization time, the initial contents of the Routing tables (namely, the list of known node addresses) and the initial values of the other Router attributes are logged.

EXCEPTION AND OTHER OCCURRENCES

Occurrences in the Router that are logged, include:

Routing Table changes,
Router attribute changes,
Errors processed by the Router.

The information is passed to the NSM via responses to commands such as ATTACH, DETACH, SET ATTRIBUTE, etc., and via reports such as FRAME ERROR REPORT.

When error cases concerning particular frames are logged, only the Router, Port, and Subport headers of the frame in error are recorded in the log. The user's text is not logged.

TRAFFIC PROFILE

The major function of Router monitoring is to generate a profile of traffic through the Router. It is anticipated that the need for this, and the kinds and amounts of information monitored, will vary widely from network to network and from time to time in a given network.

The following data for every frame is used for the Router monitoring function:

1. A copy of the Router frame in question, including: NS Header (including Router , Port , and Subport Header; the Router Header, in turn, including Destination Node Address (DNA) and Originating Node Address (ONA)), and user text.

2. The Router frame length (including NS Header).

3. Neighbor Node Address from which the Router frame was received.

4. Neighbor Node Address to which the Router frame was routed.

If Router Monitor Copy has been set (via the MONITOR command) all of this information is forwarded to the NSM for logging.

If TRAFFIC has been specified by the MONITOR command the Router accumulates a traffic profile summary, and periodically is interrogated by the NSM for logging. The time interval for this is determined by the MONITOR command option, INTERVAL.

The traffic profile information is maintained as a two dimensional matrix, DNA on one axis and ONA on the other. Each element of the matrix consists of six parts:

1. The number of Router Info Frames addressed to DNA from ONA which were routed by this node without error.

2. The average length of these frames.

3. The number of Router Control Frames addressed to DNA from ONA which were routed by this node without error.

4. The average length of these frames.

5. The number of Router Frames addressed to DNA from ONA concerning which errors were detected by this node.

6. The average length of these frames.

STATION LEVEL LOGGING AND MONITORING

Logging Functions

Due to the many events that occur on communications links to interrupt service, it is essential that these be monitored and logged very closely. To this end the Station Level is able to log a wide variety of conditions that occur on the communications medium. Each type of station has its own set of conditions that need to be logged. Stations that use telephone circuits for their communications typically provide more error types to be identified; links such as Global Memory (TM), provide fewer error types.

The events that occur that cause reports from the Station Groups may be controlled within classes. These event classes are controlled by setting Station Level attributes. These control attributes (Event-Monitor attributes) may be set using the ADD and MODIFY STATION commands.

Counts

Some of the events that occur are not individually reported, but counts of these events are accumulated in the Station Groups (depending upon the setting of the monitor control attributes). The SLM periodically interrogates these counters and sends the results to the NSM for logging.

The Station Group counts are also available to an Operations Interface agent by requesting COUNTS with the the Station Inquiry command.

The setting of Station Level Monitor attribute will enable the counters, and the MONITOR command (with INTERVAL) determines the frequency that the counts are logged.

Also when a Station is closed, and the Station is in a monitoring mode, the counts are interrogated and reported to the NSM for logging.

Figure 8-1 shows a sample of the conditions that may be logged or monitored by a BDLC Station Group:

MONITORING FUNCTION

EVENT TO BE MONITORED		COLLECTION				REPORTING		APPLICATION					REMARKS
EVENT ID SG-	EVENT	OPTIONAL	DATA CAPTURE	COUNT OCCURRENCES	TIME-STAMPED	UPON OCCURRENCE	ON DEMAND	BILLING	PERFORMANCE	MAINTENANCE	DEBUGGING	SECURITY	
1	BDLC-FRAME-RECEIVED	√		√			√		√	√	√		ALLOWS DETERMINATION OF INBOUND FRAME ERROR RATE.
2	FCS-FAILURE	√		√			√		√	√	√		
3	BDLC-FRAME-SENT	√		√			√		√	√	√		
4	I-FRAME-SENT	√		√			√		√	√	√		
5													
6													
7	REMOTE-BUSY-SET/RESET	√			√	√			√	√	√		
8	I-RSP-RECEIVED, ACI	√	√		√	√			√	√	√		PROBABLE LOOP BACK (X,25 ADDRESSING CONVENTION)
9	CONNECTION-PORT-DIALOG-OPEN/CLOSED	√			√	√		√		√	√		TO DETERMINE DURATION OF SWITCHED LINE CONNECTION
10	INVALID-BDLC-ADDRESS, ACI		√		√	√			√	√	√	√	UNKNOWN HOST ATTACHED
11	BDLC-FRAME-RECEIVED-NOT-CCTET-MULTIPLE, ACI		√		√	√			√	√	√		
12	BDLC-FRAME-TCO-SHORT, ACI		√		√	√			√	√	√		
13	UNEXPECTED-CONNECTION-PORT-DLG-CLOSED				√	√			√	√	√		ALARM
14	NON-BNA-CALLER				√	√			√	√	√	√	SEE NOTE 1.
15	LINK-RESET-LOCALLY, I.E. FRMR-RECEIVED OR UNEXPECTED-UA/DM-RECEIVED OR RETRY-COUNT-EXCEEDED				√	√			√	√	√		
16	FRMR-RECEIVED, I		√		√	√			√	√	√		
17	RETRY-COUNT-EXCEEDED				√	√			√	√	√		
18	LINK-RESET-REMOTELY				√	√			√	√	√		
19	FRMR-SENT, I		√		√	√			√	√	√		
20	UNEXPECTED-DM/DISC-RECEIVED				√	√			√	√	√		
21	STATION-DIALOG-REOPENED				√	√			√	√	√		

NOTE 1. DIAL-RSP-TIMER EXPIRES BEFORE STATION-DIALOG IS OPEN.

EB1002

Figure 8-1. BDLC Station Logging Information

SECTION 9

FRAME FORMATS

This section describes the frame format of messages in Network Services. The formats are discussed in three major groups: Link Frames, Router Frames and Port Frames. Figure 9-1 shows the hierarchy of frame formats. The area of user data flow is shown with heavy lines. These are the User or Host Services messages as they are routed through Network Services. Other frame formats shown to the left of the main path are for Station Level, Router Level, and Port Level communications and control. These are also discussed in this section. Figure 9-1 is organized in the order in which the formats are discussed in this section.

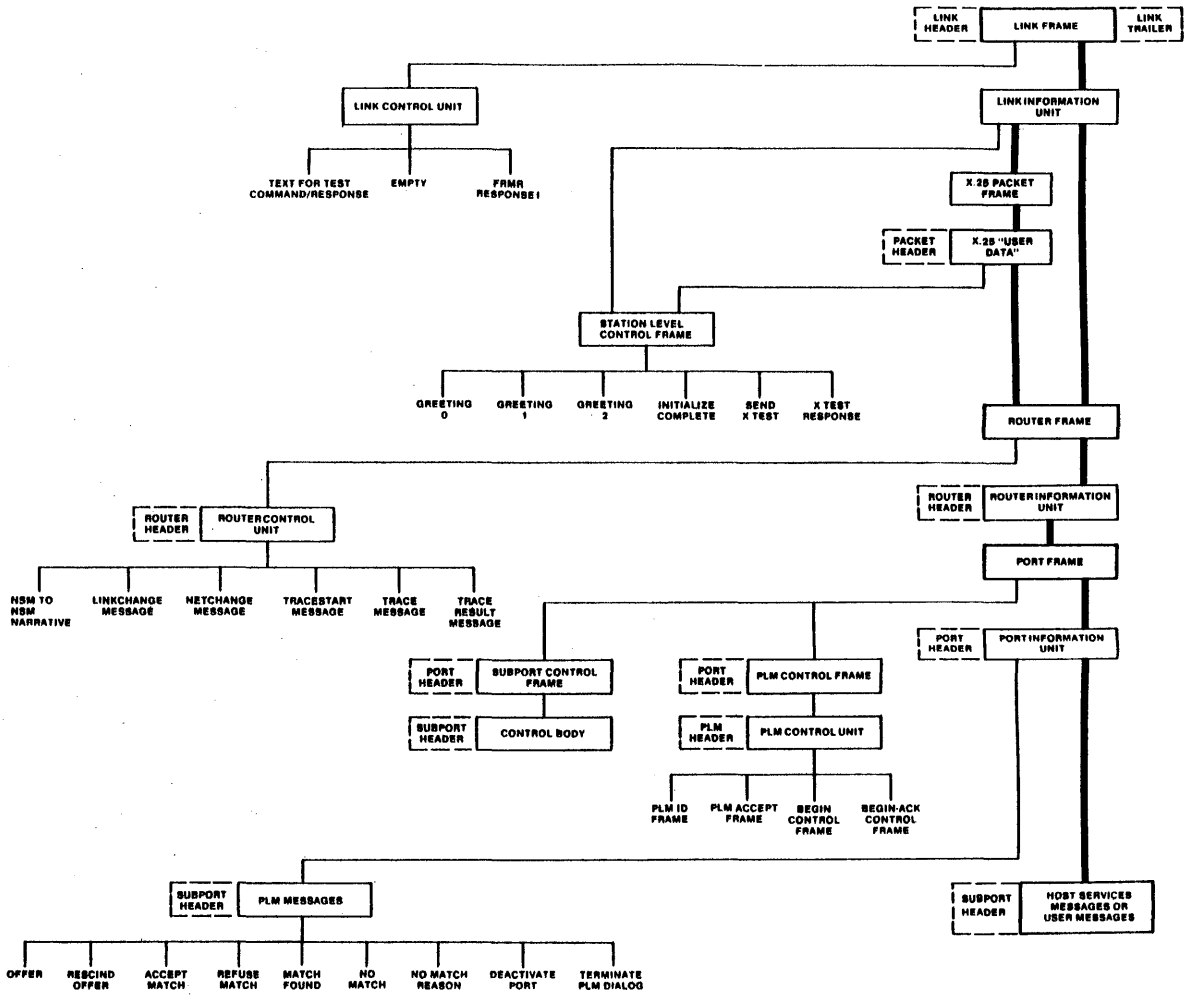
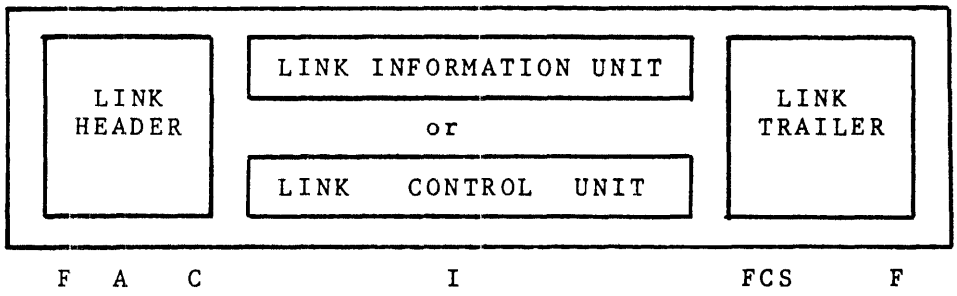


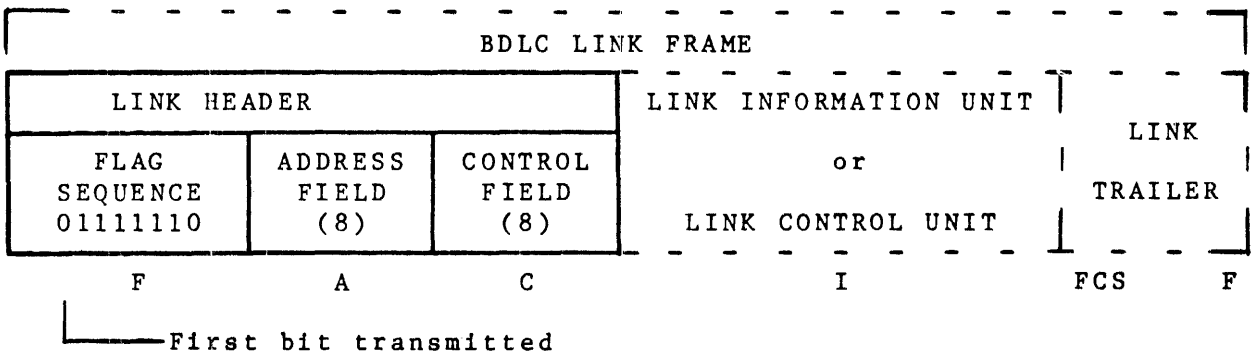
Figure 9-1. Frame Format Hierarchy

BDLC LINK FRAME



A LINK FRAME is composed of a LINK HEADER, a LINK TRAILER, and a body (I-Field) which can be either a LINK INFORMATION UNIT or a LINK CONTROL UNIT.

LINK HEADER (F, A, C)



Flag Sequence (F)

All frames start and end with the flag sequence. This sequence is a zero bit followed by six one bits followed by a zero bit (01111110). All stations attached to the data link continuously scan, on a bit-by-bit basis, for this sequence. The flag is used for frame synchronization. The flag sequence which closes a frame can also be the opening flag sequence for the next frame. Any number of complete flags can be used between frames.

Address Field (A)

Since all BNA node-to-node and X.25 links are point-to-point, only two values are used in the address field:

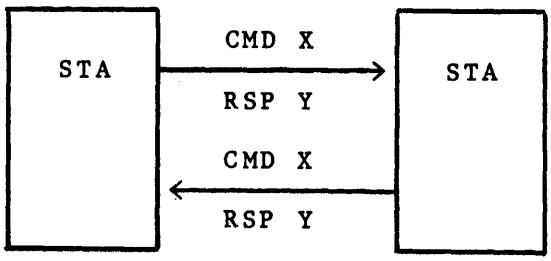
X - 1100 0000

Y - 1000 0000

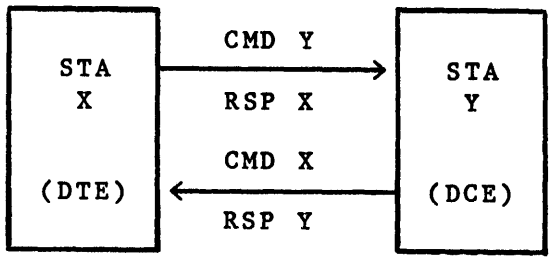
|

Least significant bit,
transmitted first

For BNA node-to-node links, the X value denotes a COMMAND and the Y value denotes a RESPONSE:



When the BDLC STATION GROUP supports LAP B in the X.25 STATION GROUP, standard BDLC addressing is used, assigning each station a different address:



Control Field (C)

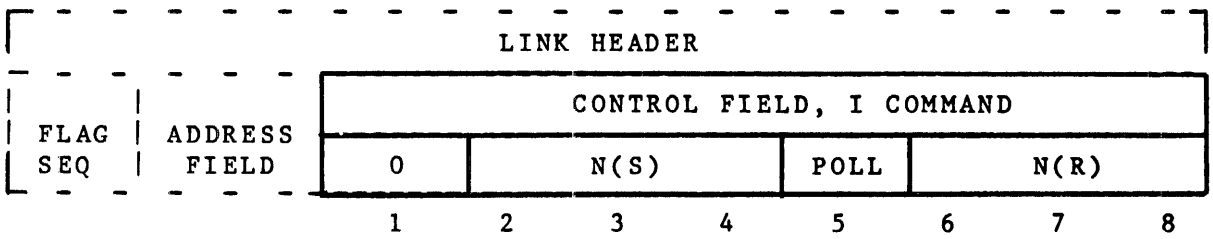
The Control Field contains a command or response and can contain sequence numbers. The control field is used by the transmitting station to tell the addressed station what operation it is to perform. It is also used in the receiving stations response. The length of this field is eight bits. There are three categories of commands and responses:

I Command, Information Transfer Format
S Commands and Responses, Supervisory Format
U Commands and Responses, Unnumbered Format

I COMMAND, INFORMATION TRANSFER FORMAT

Transfers sequentially numbered frames containing an optional information field across the link. Data is transferred only in command I frames, never in response frames.

Control Field, I Command



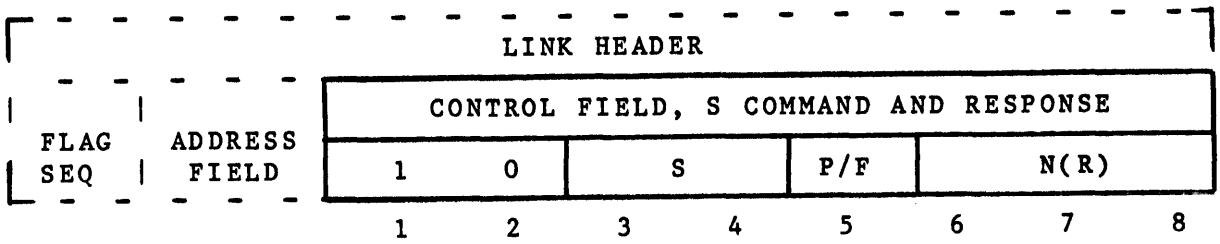
Bit	Description
1	Always 0. Identifies the Information Transfer Format.
2-4	N(s) - SEND SEQUENCE NUMBER, Modulo 8 Indicates the sequence number of the I frame.
5	POLL. When set to 1, indicates that a response is required.
6-8	N(R) - RECEIVE SEQUENCE NUMBER, Modulo 8. Indicates the sequence number of the next expected I frame (that is, I frames numbered up to and including N(R)-1 are accepted).

The LSB of each field is transmitted first.

S COMMANDS AND RESPONSES, SUPERVISORY FORMAT

Supervisory, (S) commands and responses are used to perform basic supervisory link control functions such as I frame acknowledgment, polling, temporary interruption of information transfer, and error recovery. Frames with the S format do not contain an information field.

Control Field, S Commands and Responses



Bit	Description
1-2	Always "1 0". Identifies the Supervisory Format.
3-4	S Field

00 - RR (Receive Ready)

Used by a station to indicate it is ready to receive an I frame and to acknowledge I frames numbered up to and including N(R)-1.

A station uses the RR command with the POLL bit set to "1" to solicit responses from a remote station.

An RR frame is one way to report the end of a station busy condition.

10 - RNR (Receive Not Ready)

Used by a station to indicate a "busy" condition: that is, the temporary inability to accept additional incoming information frames. I frames numbered up to and including N(R)-1 are acknowledged. I frame N(R) and any subsequent I frames received, if any, are not acknowledged; the acceptance status of these frames is indicated in subsequent exchanges.

A station receiving an RNR frame when in the process of transmitting is to stop transmitting at the earliest possible time by completing or aborting the frame in process.

A station uses the RNR command with the POLL bit set to "1" to obtain the receive status of the remote station.

01 - REJ (Reject)

Used by a receiving station to request retransmission of I frames before the transmitter could detect the need to retransmit by check pointing (that is, the exchange of P/F bits). I frames numbered N(R)-1 and below are acknowledged. Additional I frames pending initial transmission can be transmitted following the retransmitted I frame(s).

Only one REJ exception condition, from a given station to another station, can be established at any given time. Another REJ cannot be transmitted (that is, actioned) until the first REJ exception condition has been cleared at the sender.

The REJ exception condition is cleared (reset) upon acceptance of an I frame with an N(S) number equal to the N(R) of the REJ command/response. REJ is one way to report the end of a station busy condition.

5 P/F

Command-POLL.
When set to 1, indicates that a response is required.

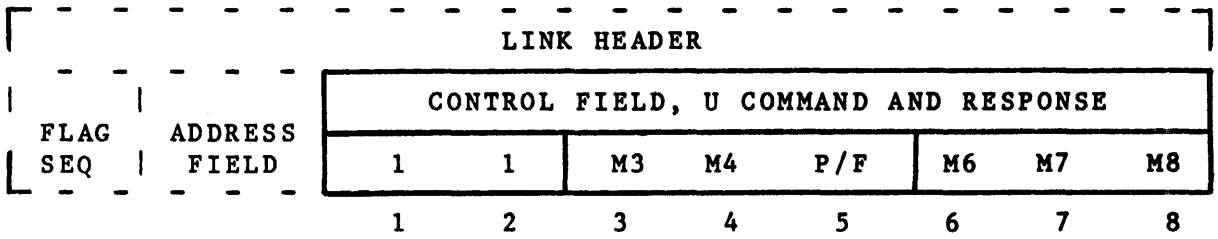
Response-Final.
When set to 1, indicates a response to a poll.

6-8 N(R)
Same as I Command

U COMMANDS AND RESPONSES, UNNUMBERED FORMAT

Unnumbered format Commands and Responses are used to extend the number of link Supervisory functions. Except as noted below, U frames do not contain an I field. Five "modifier" bits are used to identify the additional command and response functions.

Control Field, U Commands and Responses



Bits	Description
1-2	Always "1 1". Identifies the Unnumbered format.
3-4	Modifier Bits. See bit 6-8 below.
5	P/F. Same as S Command
6-8	Modifier Bits

3	4	6	7	8	COMMAND	RESPONSE
1	1	1	0	0	SABM	
0	0	0	0	0	DISC	
0	0	1	1	1	TEST	TEST
0	0	1	1	0		UA
1	1	0	0	0		DM
1	0	0	0	1		FRMR

SABM - Set Asynchronous Balanced Mode (ABM) Command

The SABM command places the addressed balanced station in ABM. ABM is the operational BDLC mode used in a BNA BDLC station. In this mode, the stations have identical data transfer and link control capability. A station may initiate transmission without receiving permission from the other station.

SABM always has P=1. Upon acceptance of this command the receiving station SEND and RECEIVE VARIABLEs are set to zero. The receiving station confirms acceptance of SABM by the transmission of a UA response.

Previously transmitted I frames that are unacknowledged when this command is actioned remain unacknowledged. Transmission of SABM is one way to report the end of a Balanced station busy condition.

DISC - Disconnect Command

The DISC command performs a logical disconnect; i.e., inform the addressed station that the transmitting station is suspending operation. In switched networks, this logical disconnect function at the data link level initiates a physical disconnect operation at the physical interface level; i.e., to go "on-hook". The DISC command always has P=1.

The receiving station confirms acceptance of DISC by the transmission of a UA response. A station in Asynchronous Disconnect Mode (ADM) will transmit a DM response upon receiving a DISC command.

Previously transmitted I frames that are unacknowledged when this command is actioned remain unacknowledged.

TEST - Test Command

The TEST command causes the remote station to return a TEST response at the first respond opportunity. TEST always has P=1.

An information field is optional with the TEST Command. If present, the information field is returned by the remote station with the TEST Response.

The station considers the test terminated upon receipt of the TEST Response or when a timer has expired. The results of the TEST Command/Response are made available for interrogation by a higher level.

TEST - Test Response

The TEST Response replies to the TEST Command. An information field is optional with the TEST Command and, if present, is returned with the corresponding TEST Response.

A station in any mode receiving a TEST Command will transmit a TEST Response unless a set mode response (UA) is pending, or a Frame Reject Response (FRMR) condition exists.

A FRMR condition will not be established if the received TEST command has an information field which exceeds the maximum established storage capability of the Secondary/Balanced station.

UA - Unnumbered Acknowledgment Response

Used to acknowledge the receipt and acceptance of the SABM and DISC commands.

DM - Disconnected Mode Response

Used to report a non-operational status where the station is logically disconnected from the link, that is, the station is in ADM (Asynchronous Disconnect Mode). The DM response is sent by a station to inform the remote station that the local station is still in ADM and cannot action the mode setting command.

FRMR - Frame Reject Response

Used to report an error condition not recoverable by retransmission of the identical frame; i.e., one of the following conditions resulted from the receipt of an error-free frame from the remote station.

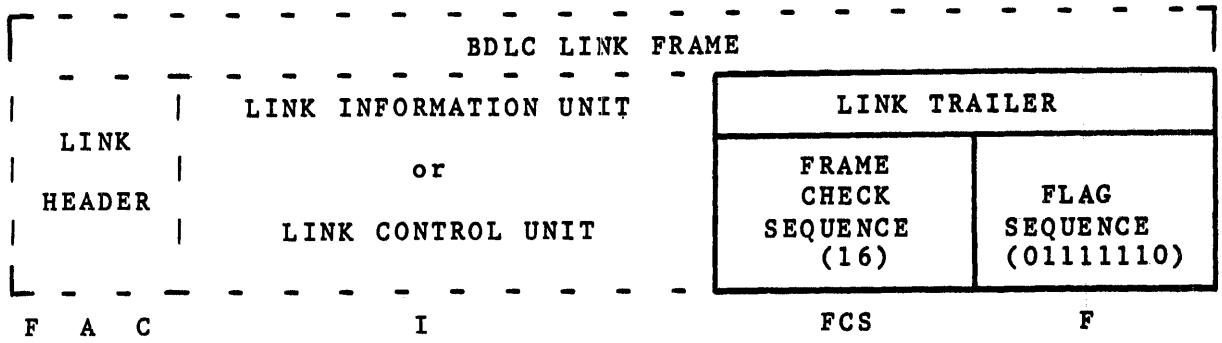
1. The receipt of a frame that is invalid or not implemented.

2. The receipt of a frame with an information field which exceeded the maximum established length.

3. The receipt of an invalid N(R) number from the remote station. An invalid N(R) is defined as a number which points to an I frame which has previously been transmitted and is not the next sequential I frame pending transmission.

An I field is returned with this response to provide the reason for the Frame Reject Response.

LINK TRAILER (FCS, F)



Frame Check Sequence (FCS)

All frames include a 16-bit frame check sequence (FCS) just prior to the closing flag for error detection purposes. The contents of the ADDRESS (A), CONTROL (C) and INFORMATION (I) fields, excluding the zeros inserted to maintain transparency are included in the calculation of the FCS sequence.

The FCS is the remainder of a module 2 division process utilizing a generator polynomial as a divisor. The generator polynomial is that used in CCITT Recommendation V.41 and is:

$$x^{16} + x^{12} + x^{5} + 1$$

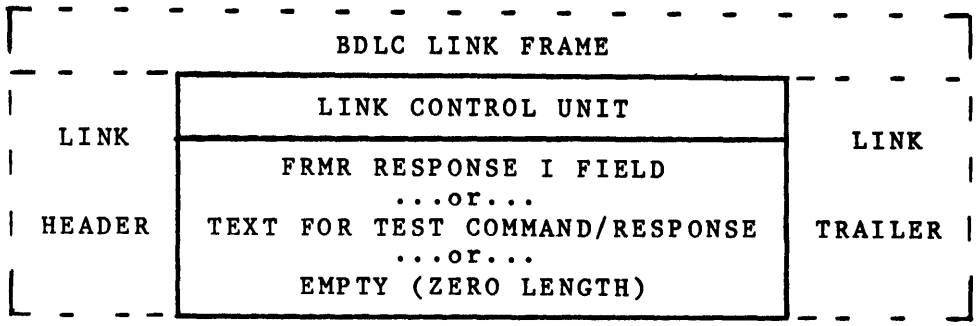
The order of bit transmission is least significant bit first.

Flag Sequence (F)

All frames start and end with the flag sequence. This sequence is a zero bit followed by six one bits followed by a zero bit (01111110). All stations attached to the data link continuously scan, on a bit-by-bit basis, for this sequence. The flag is used for frame synchronization. The flag sequence which closes a frame can also be the opening flag sequence for the next frame. Any number of complete flags can be used between frames.

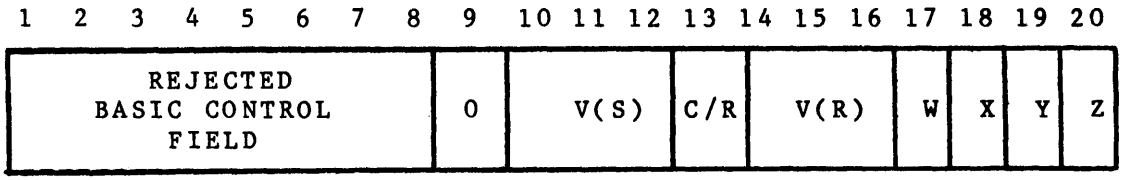
LINK CONTROL UNIT

The INFORMATION (I) field or body of the LINK FRAME is a LINK CONTROL UNIT when the value of Bit 1 of the CONTROL (C) field in the LINK HEADER is a 1, indicating either the Supervisory (S) or Unnumbered (U) format.



The LINK CONTROL UNIT can be the I field of FRMR Unnumbered Responses interpreted by the BDLC STATION or the optional I field of TEST Unnumbered Commands and Responses, text transferred to/from the operator via the SLM. The length of the LINK CONTROL UNIT is zero for Supervisory (S) Format Commands and Responses, and for the SABM, DISC, UA, and DM Unnumbered (U) Commands and Responses.

FRAME REJECT RESPONSE (FRMR) I FIELD



Bits	Description
1-8	The control field of the received frame which causes the reject.
9	Always 0.
10-12	V(S). The current SEND VARIABLE S number of the station detecting the FRMR condition.
13	C/R If zero, indicates a command is being rejected. If one, indicates a response is being rejected.

14-16 V(S)
The current RECEIVE VARIABLE S number of the station detecting the FRMR condition.

17 W.
If "1" indicates that the control field received and returned in bits 1-8 was invalid or not implemented.

18 X.
If "1" indicates the control field received and returned in bits 1 through 8 was considered invalid because the frame contained an information field which is not permitted with this command. Bit W must be set to "1" in conjunction with this bit.

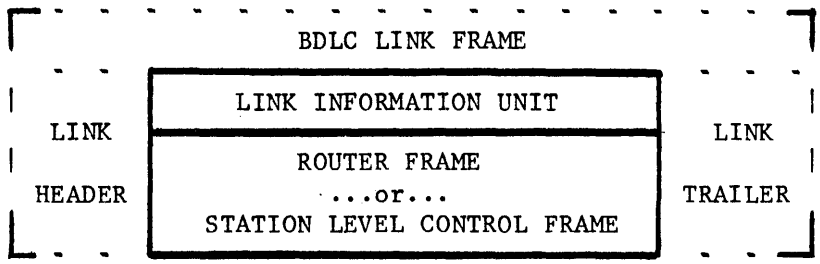
19 Y.
If "1" indicates the information field received exceeded the maximum established capability of the station.

20 Z.
If "1" indicates the control field received and returned in bits 1 through 8 contained an invalid N(R) number.

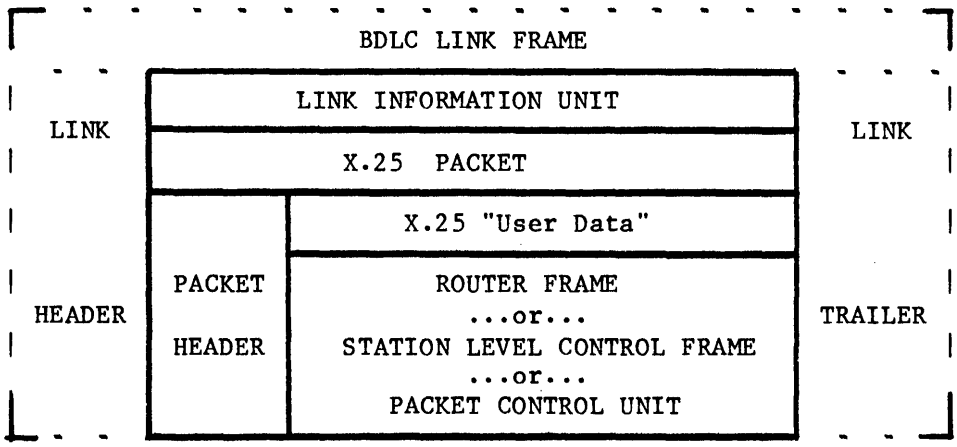
The I field associated with the FRMR is padded with 4 zero bits so as to end on a byte boundary. FRMR can have bits W, X, Y, and Z all set to 0.

LINK INFORMATION UNIT

The INFORMATION (I) FIELD or body of the LINK FRAME is a LINK INFORMATION UNIT when Bit 1 of the CONTROL (C) FIELD in the LINK HEADER is a 0, indicating the Information Transfer format.



If the STATION GROUP is X.25, an additional level of interpretation exists - the PACKET level - in which the LINK INFORMATION UNIT is the X.25 PACKET.



The value of the LEVEL subfield of the FRAME TYPE (FTY) field, the first field in the LINK INFORMATION UNIT, (or X.25 "User Data") identifies the LINK INFORMATION UNIT (or X.25 "User Data") as a ROUTER FRAME or a STATION LEVEL CONTROL FRAME. When the value of the LEVEL subfield is a 0, the LINK INFORMATION UNIT (or X.25 "User Data") is a ROUTER FRAME and is transferred between the STATION and the ROUTER. When the value of the LEVEL subfield is a 1, the LINK INFORMATION UNIT (or X.25 "User Data") is a STATION LEVEL CONTROL FRAME and is transferred between the STATION and the STATION LEVEL MANAGER.

MISCELLANEOUS LINK INFORMATION

Abort

Abort is the procedure by which a station in the process of sending a frame, ends the frame in an unusual manner such that the receiving station will ignore the frame.

Aborting a frame is performed by transmitting at least seven, but less than fifteen, contiguous one bits (with no inserted zeros). Receipt of seven contiguous one bits is interpreted as an Abort. Receipt of fifteen or more contiguous one bits is interpreted as an Abort and Idle Link State.

Transparency

BDLC provides transparency for data coded in the information field. The occurrence of a sequence of bits that look like the flag sequence in the ADDRESS, CONTROL, INFO and FCS fields is prevented via a "zero bit insertion" procedure as follows:

The transmitter inserts a zero bit following five contiguous one bits anywhere between the beginning flag and the ending flag of the frame. The insertion of the zero bit thus applies to the ADDRESS, CONTROL, INFO and FCS fields, including the last 5 bits of the FCS. The receiver continuously monitors the received bit stream. Upon receiving a zero bit followed by five contiguous one bits, the receiver inspects the following bit. If it is a zero, the 5 one bits are passed as data and the zero bit deleted. If the sixth bit is a one, the receiver inspects the seventh bit. If the seventh bit is a zero, a flag sequence has been received. If it is a one an abort has been received.

Active Link State and Interframe Time Fill

A link is in an Active state when a station is actively transmitting a frame, an abort sequence, or interframe time fill. When the link is in the active state the right of the transmitting station to continue transmission is reserved.

Interframe time fill is accomplished by transmitting continuous flags between frames. There is no provision for intraframe time fill.

Idle Link State

A link is in an Idle state when a continuous ones state is detected that persists for at least 15 bit times.

Idle link time fill is a continuous one condition on the link.

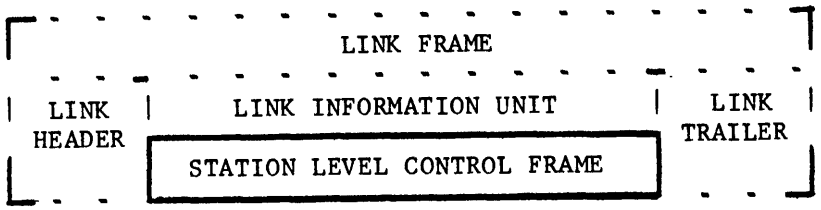
Invalid Frame

An invalid frame is one that is not properly bounded by two flags (thus an aborted frame is an invalid frame) or one which is too short (that is, shorter than 32 bits between flags). A station ignores any invalid frame.

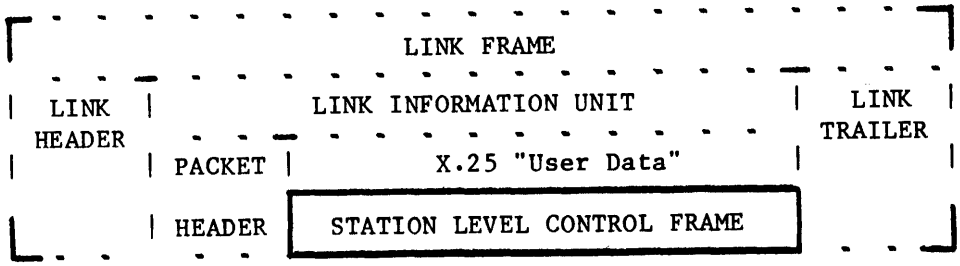
STATION LEVEL FRAME

STATION LEVEL CONTROL FRAMES

When the value of the LEVEL subfield of the FRAME TYPE field (the first field in the LINK INFORMATION UNIT) is 1, the LINK INFORMATION UNIT (or X.25 "User Data") is a Station Level Control Frame.



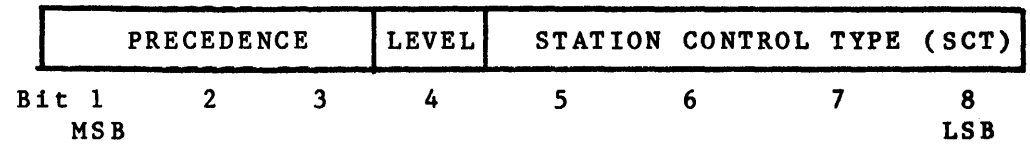
...or...



There are six Station Level control frames: GREETING 0, GREETING 1, GREETING 2, STATION LEVEL INITIALIZATION COMPLETE, SEND X TEST and X TEST RESPONSE.

Frame Type

The first byte of each frame, called FTY (FRAME TYPE) identifies the frame. The format of the FTY is as follows:



PRECEDENCE

Field Length: 3 bits (bit 1 msb thru bit 3)
Coding: 1 octal digit
Values: 0 = ROUTINE
        1 = PRIORITY
        All other values reserved
Use: Always "1" for a Station Level control frame.

LEVEL

Field Length: 1 bit (bit 4)
Coding: 1 bit
Values: 0 = ROUTER LEVEL
        1 = STATION LEVEL
Use: Always "1" for a Station Level control frame. If "0", the frame is a ROUTER FRAME and the SCT field below has a different meaning.

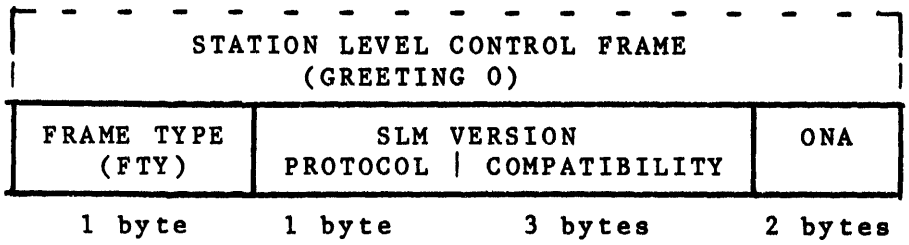
STATION CONTROL TYPE = SCT

Field Length: 4 bits (bits 5 thru 8 lsb)
Coding: 1 hex digit
Values: 0 = invalid
        1 = GREETING 1
        2 = GREETING 2
        3 = INIT COMPLETE
        4 = SEND X TEST   (X.25 STATION GROUP only)
        5 = X TEST RESPONSE   (X.25 STATION GROUP only)
        6 = GREETING 0
        All other values reserved
Use: When LEVEL = STATION LEVEL, indicates the type of Station Level Control Frame.

Station Level Greetings

Station Level Greeting messages are used in the Station Level neighbor node validation and authentication processes. A Greetings interchange is performed by each end of a link when bringing a link up and after any link reset operation. The GREETING 0 message identifies its sender. The GREETING 1 message contains parameters which define the Station Level connection and requests permission to send traffic. The GREETING 2 message, after validation of the Greeting 0 and Greeting 1, grants (or denies) permission to send traffic.

GREETING 0



Frame type:

Field Length: 8 bits in 3 subfields
Values: PRECEDENCE = 001 = PRIORITY
LEVEL = 1 = STATION LEVEL
STATION CONTROL TYPE = 0110 = GREETING 0

SLM VERSION: Consists of two sub-fields:
PROTOCOL
COMPATIBILITY

PROTOCOL

Field Length: 1 byte (8 bits)
Coding: 8 bit number
Use: Identifies present protocol level of SLM.

COMPATIBILITY

Field Length: 3 bytes (24 bits)
Coding: Bit mask
Use: This field specifies the prior protocol levels that the SLM supports. The first bit (msb) identifies the version prior to the present one, the second most significant bit identifies the version prior to that, etc. If a bit = ONE, the SLM supports the identified protocol version.

ONA - Origin Node Address

Field Length: 16 bits
Coding: Arbitrary, except
all zeros =invalid
all ones = reserved
Values: Each node in a network has a unique node address

GREETING 1

STATION LEVEL CONTROL FRAME (GREETING 1)

FRAME TYPE (FTY)	ONA	NMSS	LMSS	INIT	PASS WORD	NW VERSION

Frame type:

Field Length: 8 bits in 3 subfields
Values: PRECEDENCE = 001 = PRIORITY
LEVEL = 1 = STATION LEVEL
STATION CONTROL TYPE = 0001 = GREETING 1

ONA - Origin Node Address

Field Length: 16 bits
Coding: Arbitrary, except
all zeros =invalid
all ones = reserved
Values: Each node has a unique node address
Use: Identifies the node sending the frame. An exact match to the ONA in Greeting 0 is required.

NETWORK MAX SEGMENT SIZE = NMSS

Field Length: 2 bytes (16 bits)
Coding: 16 bit integer
Values: 60....65535
Use: This field indicates the sending node's NMSS. It is used to verify that all nodes in a network have a common NMSS.

LINK MAX SEGMENT SIZE = LMSS

Field Length: 2 bytes (16 bits)
Coding: 16 bit integer
Values: 60....65535
Use: This field indicates the sending node's LMSS. It is used to establish the working LMSS for the station dialog.

INITIALIZING INDICATOR = INIT

Field Length: 1 byte
Coding: Integer
Values: 0 = invalid
1 = INITIALIZING
2 = OPERATING
All other values are reserved
Use: INIT = INITIALIZING indicates that the sending SLM is in initialization phase 3. INIT = OPERATING indicates that the sending SLM is in normal operation phase.

PASSWORD

The PASSWORD consists of two subfields:
PASSWORD LENGTH
MY PASSWORD TO REMOTE SLM

PASSWORD LENGTH

Field Length: 1 byte
Coding: Binary integer
Values: 0 - 17
Use: This field gives the length in bytes of the MY PASSWORD TO REMOTE SLM field. This length field must always be present. Zero means null, no password.

MY PASSWORD TO REMOTE SLM

Field Length: 0 to 17 bytes
Coding: Ebcdic string
Values: Arbitrary
Use: This field contains MY PASSWORD TO REMOTE SLM and is used by a remote node to verify the identity of the local node. If the receiving node has a password for the sending node, an exact match is required for the connection to be made. If the receiving node has no password for the sending node, any password is accepted.

NETWORK VERSION

The NETWORK VERSION consists of two subfields:
NETWORK VERSION LENGTH
NETWORK VERSION VALUE

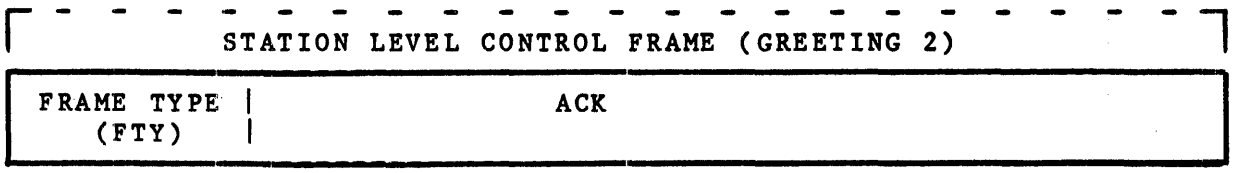
NETWORK VERSION LENGTH

Field Length: 1 byte
Coding: Binary integer
Values: 0 - 17
Use: This field gives the length in bytes of the NETWORK VERSION VALUE field. This length field must always be present. An exact match is required for the connection to be made. Zero means null, no value.

NETWORK VERSION VALUE

Field Length: 0 to 17 bytes
Coding: Ebcdic string
Values: Arbitrary
Use: This field contains NETWORK VERSION, which can be used by a network administrator to control connection of a node. An exact match is required for the connection to be made, even if zero length. Null matches null.

GREETING 2



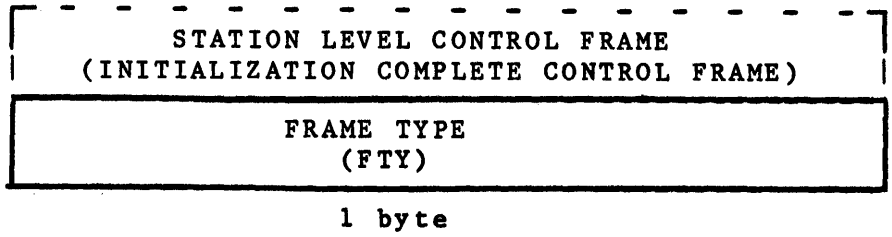
Frame type:

Field Length: 8 bits in 3 subfields
Values: PRECEDENCE = 001 = PRIORITY
LEVEL = 1 = STATION LEVEL
STATION CONTROL TYPE = 0010 = GREETING 2

ACK

Field Length: 1 byte
Coding: Integer
Values: 0 = invalid
1 = ACK
2 = STATION LEVEL VERSION NOT RESOLVED
3 = NETWORK VERSION MISMATCH
4 = NMSS MISMATCH
5 = NODE ADDRESS VALIDITY FAILURE
6 = SHUTDOWN IN PROGRESS
7 = PASSWORD MISMATCH
8 = INSUFFICIENT RESOURCES
9 = INVALID GREETING FORMAT
all other values reserved
Use: This field indicates the results of the validation process at the sending node. ACK means permission is granted to send traffic. Values 2 through 9 mean permission to send traffic is denied and implies that the line will be disconnected (if possible).

INITIALIZATION COMPLETE CONTROL FRAME



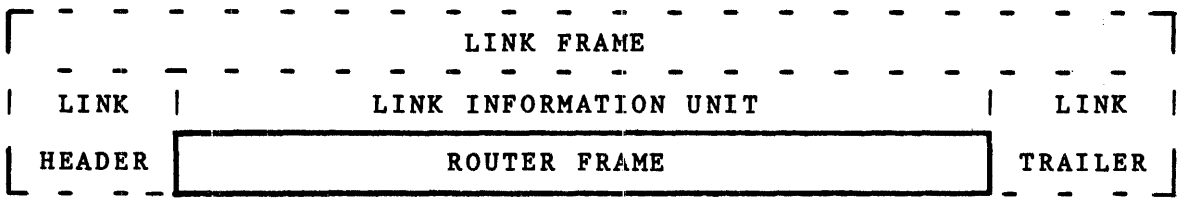
The INIT COMPLETE frame consists of one byte, FTY. It indicates that the sending node has completed Station Level initialization, which then causes the NEIGHBOR STATUS (at the receiving node) to be set to ATTACHED.

Frame type:

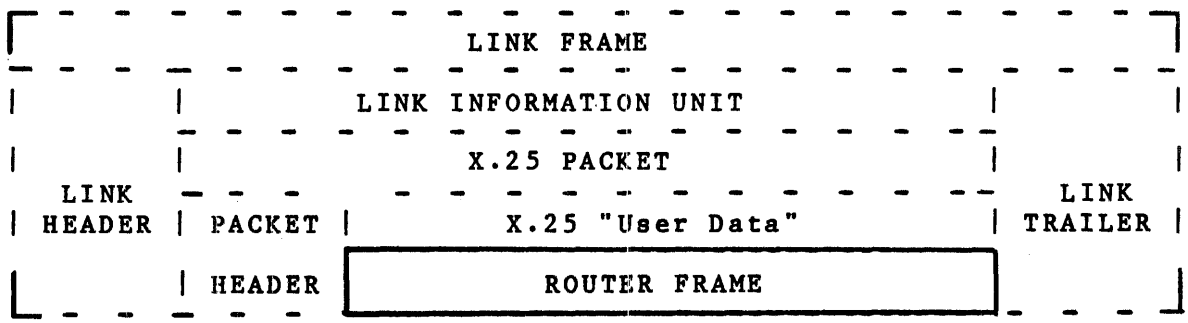
Field Length: 8 bits in 3 subfields
Values: PRECEDENCE = 001 = PRIORITY
LEVEL = 1 = STATION LEVEL
STATION CONTROL TYPE = 0011 = INITIALIZATION COMPLETE

ROUTER FRAME

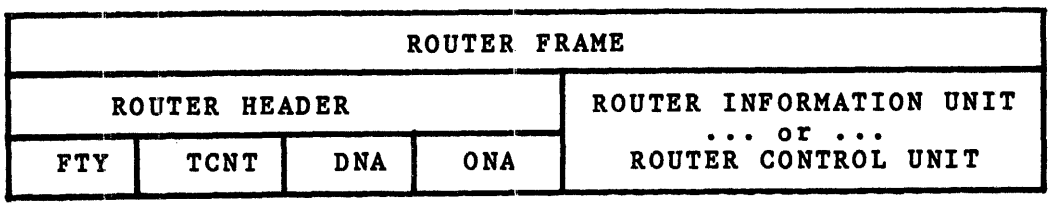
When the STATION GROUP is BDLC, the ROUTER FRAME is the LINK INFORMATION UNIT. When the STATION GROUP is X.25, the ROUTER FRAME is the X.25 "User Data", and the X.25 "User Data" is the body of the X.25 PACKET, which is the LINK INFORMATION UNIT.



...OR...

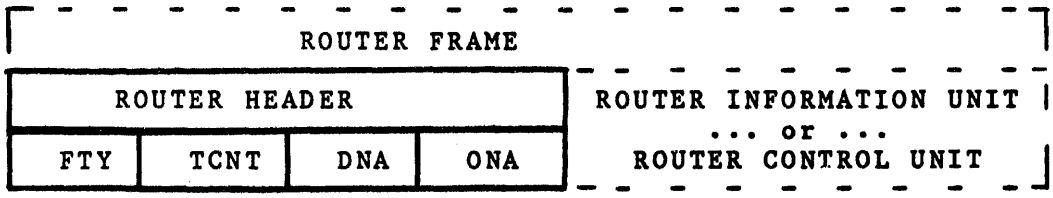


The ROUTER FRAME consists of a ROUTER HEADER and a body. The ROUTER FRAME body can be either a ROUTER INFORMATION UNIT or a ROUTER CONTROL UNIT.



ROUTER HEADER

The ROUTER HEADER consists of four fields: FRAME TYPE, TRANSIT COUNT, DESTINATION NODE ADDRESS, and ORIGIN NODE ADDRESS.



Frame Type (FTY)

Field Length: 1 byte
Coding: Consists of three sub-fields:
PRECEDENCE = bit 7 (msb) through bit 5
LEVEL = bit 4
ROUTER FRAME TYPE = bit 3 through bit 0 (lsb)

PRECEDENCE

Field Length: 3 bits
Coding: 1 octal digit
Values: 0 = ROUTINE
1 = PRIORITY
All other values reserved
Use: Indicates the precedence of the frame. PRIORITY frames are transmitted ahead of ROUTINE frames.

LEVEL

Field Length: 1 bit
Coding: 1 binary bit
Values: 0 = ROUTER LEVEL
1 = STATION LEVEL
Use: Always "0" for a ROUTER frame (including normal user traffic). If "1", the frame is a Station Level control frame and the RFT field below has a different meaning.

ROUTER FRAME TYPE = RFT

Field Length: 4 bits
Coding: 1 hex digit
Values: 0 = invalid
1 = INFORMATION
2 = CONTROL
All other values reserved

Use: When LEVEL = ROUTER LEVEL, indicates whether the body of the ROUTER FRAME is a ROUTER CONTROL UNIT (RFT = CONTROL) or a ROUTER INFORMATION UNIT (RFT = INFORMATION).

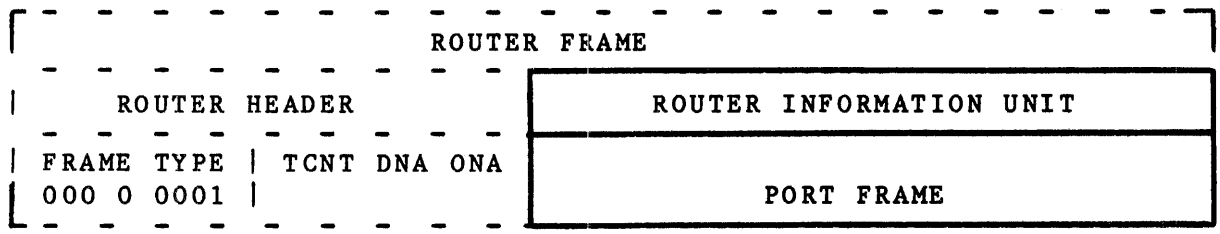
Transit Count (TCNT)

Field Length: 1 byte
Coding: B binary number
Values: Non-negative integers (0..255)
Use: This field is used by the ROUTER to detect ROUTER FRAMES which are looping in the network. When a Frame is received at a node, the TCNT equals the number of links it has traversed.

Destination Node Addr. Origin Node Addr (DNA, ONA)

Field Length: 2 bytes each
Coding: Arbitrary, except:
all zeros = invalid
all ones = (reserved)
Values: Each node in a network must have a unique node address in that network.
Use: The DESTINATION NODE ADDRESS references the node to which the ROUTER FRAME is to be delivered. The ORIGIN NODE ADDRESS references the node which originated the ROUTER FRAME.

ROUTER INFORMATION UNIT

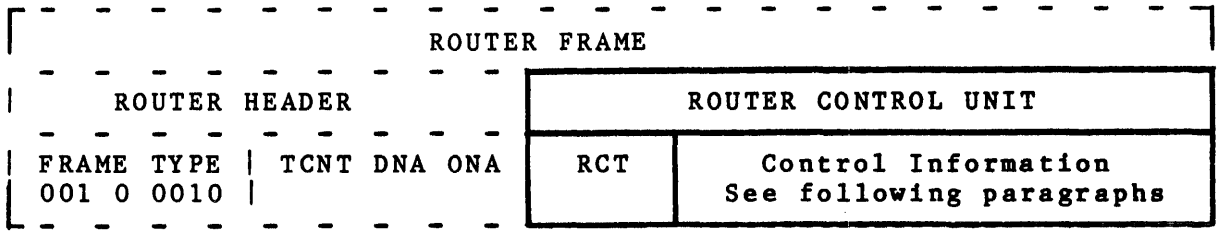


The FRAME TYPE (FTY) in the ROUTER HEADER of all ROUTER FRAMEs containing a ROUTER INFORMATION UNIT is the same:

PRECEDENCE = ROUTINE = 000
LEVEL = ROUTER = 0
ROUTER FRAME TYPE = INFORMATION = 0001

The ROUTER INFORMATION UNIT is the PORT FRAME.

ROUTER CONTROL UNIT



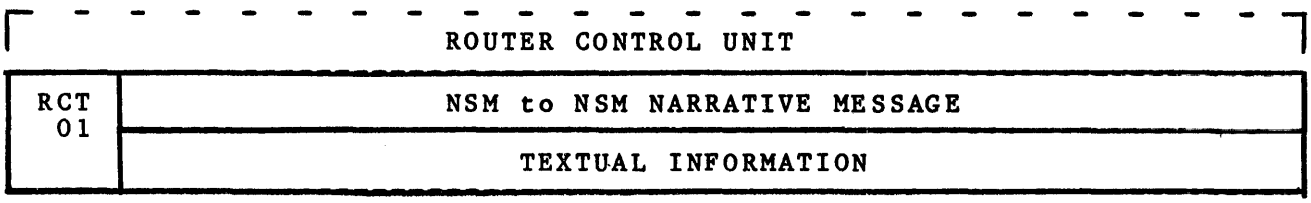
The FRAME TYPE (FTY) in the ROUTER HEADER of all ROUTER FRAMEs containing a ROUTER CONTROL UNIT is the same:

PRECEDENCE = PRIORITY = 001
LEVEL = ROUTER = 0
ROUTER FRAME TYPE = CONTROL = 0010

RCT (ROUTER CONTROL TYPE)

Field Length: 1 byte
Coding: 2 hex digits
Values: 00 = Invalid
01 = NSM-NSM NARRATIVE
02 = LINKCHANGE
03 = NETCHANGE
04 = TRACE START
05 = TRACE
06 = TRACE RESULT
All other values reserved
Use: This field identifies the type of ROUTER CONTROL FRAME.

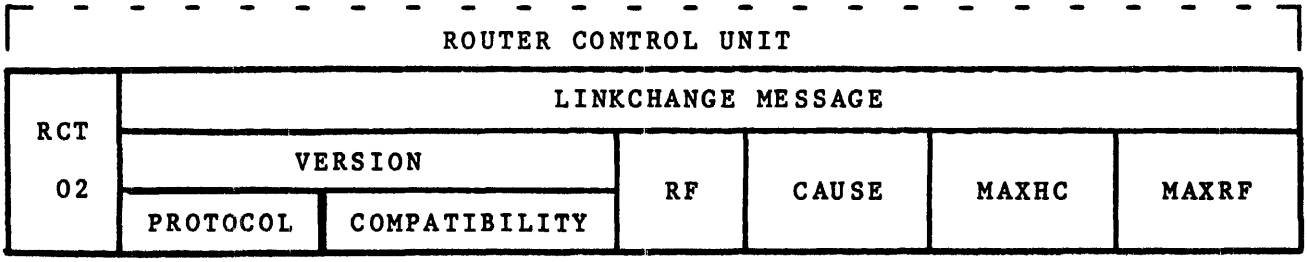
NSM to NSM Narrative



RCT = 01 (NSM NSM NARRATIVE)

NSM to NSM textual information -
The narrative information to be communicated from one NSM to another. For example, a narrative message from one human operator to another.

Linkchange Message



RCT = 02 (LINKCHANGE MESSAGE)

VERSION

Field Length: 2 subfields
ROUTER VERSION PROTOCOL
ROUTER VERSION COMPATIBILITY

ROUTER VERSION PROTOCOL

Field Length: 1 byte
Coding: Integer
Values: 1..255
Use: This field defines the BNA standard protocol level used by this version of the ROUTER.

ROUTER VERSION COMPATIBILITY

Field Length: 3 bytes
Coding: 24 bit mask
Values: The high order bit of the mask corresponds to the protocol level "(ROUTER VERSION PROTOCOL) - 1", and the low order bit corresponds to "(ROUTER VERSION PROTOCOL) - 24".
Use: This field is used by the ROUTER to verify that the nodes on each end of this logical link are using compatible versions of the ROUTER.

RF = RESISTANCE FACTOR

Field Length: 2 bytes
Coding: Integer
Values: Any
Use: This is the ONA's link resistance of the logical link between the ONA and DNA. All multiple parallel links are included in this number.

CAUSE

Field Length: 1 byte
Coding: Integer
Values: 00 = invalid
01 = TOPOLOGICAL CHANGE (attach or detach)
02 = OPERATIONS CHANGE
03 = ROUTING REFRESH
04 = ROUTING REFRESH RESPONSE
All other values are reserved
Use: This field describes the reason for the LINKCHANGE. Among other things, it resolves a potential race condition between the arrival of the ATTACH or DETACH commands and the arrival of the LINKCHANGE.

MAXHC = MAXIMUM HOP COUNT

Field Length: 1 byte
Coding: Integer
Values: Any, greater than 0
Use: This is the network maximum hop-count according to the ONA.

MAXRF = MAXIMUM RESISTANCE FACTOR

Field Length: 2 bytes
Coding: Integer
Values: Any, greater than 0
Use: This is the network maximum resistance factor according to the ONA.

Netchange Message

ROUTER CONTROL UNIT

RCT	NETCHANGE MESSAGE								
03	N	NA(1)	REA(1)	HC(1)	RF(1)	NA(n)	REA(n)	HC(n)	RF(n)

RCT = 03 (NETCHANGE MESSAGE)

N = number of NA, REA, HC, RF sets in the message

Field Length: 1 byte
Coding: Integer
Use: This number indicates the number of subject node addresses that are contained in the NETCHANGE.

NA(1..N) = NODE ADDRESS

Field Length: 2 bytes
Values: A node address
Use: This is the address of the node which is the subject of this part of the NETCHANGE.

REA(1..N) = REACHABLE

Field Length: 1 byte
Coding: Integer
Values: 00 = UNREACHABLE
01 = REACHABLE
All other values are reserved
Use: This indicates to the DNA whether or not the ONA considers the NA to be reachable.

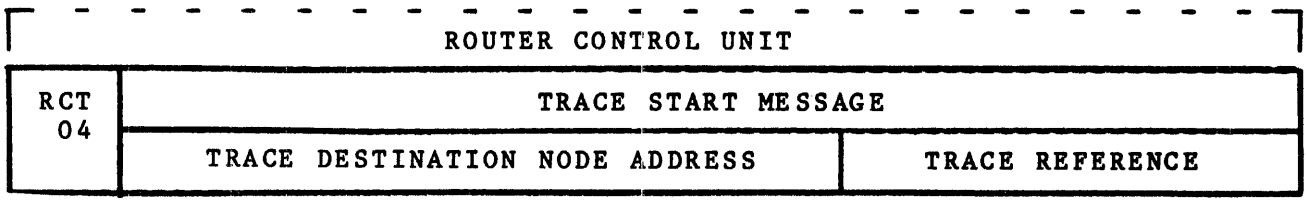
HC(1..N) = hop-count

Field Length: 1 byte
Coding: Integer
Values: Any
Use: This indicates the minimum hop-count from the ONA to the NA.

RF(1..N) = RESISTANCE FACTOR

Field Length: 2 bytes
Coding: Integer
Values: Any
Use: This indicates the minimum resistance factor from the ONA to the NA.

Trace Start



RCT = 04 (TRACE START MESSAGE)

(Note: The DNA is the Trace Source Node Address.
The ONA is the address of the initiating and receiving node).

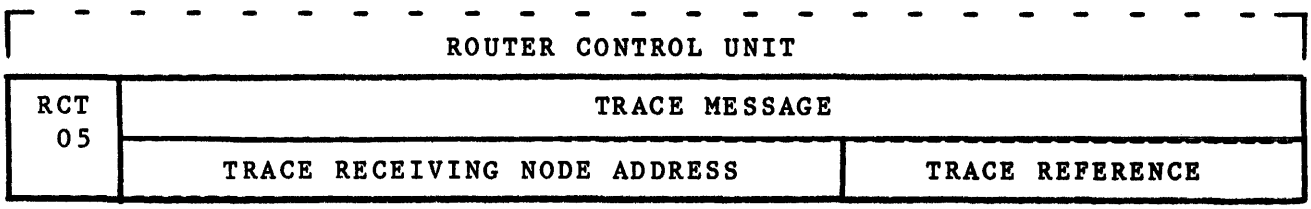
TRACE DESTINATION NODE ADDRESS

Field Length: 2 bytes
Values: A node address
Use: Specifies the destination of the resultant TRACE message.

TRACE REFERENCE

Field Length: 2 bytes
Values: Any
Use: The initiating node supplies this value to aid in the processing of TRACE RESULTs.

Trace



RCT = 05 (TRACE MESSAGE)

(Note: The DNA is the trace destination node address and the ONA is the trace source node address).

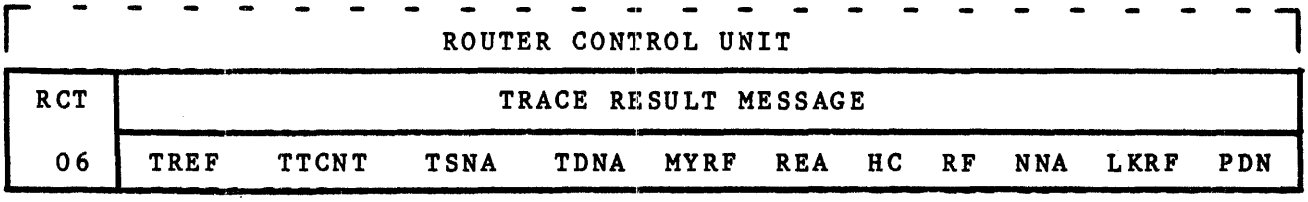
TRACE RECEIVING NODE ADDRESS

Field Length: 2 bytes
Values: A node address
Use: Specifies the destination of the resultant TRACE RESULT messages.

TRACE REFERENCE

Field Length: 2 bytes
Values: Any
Use: This value is included in the TRACE RESULT messages to aid the initiating node in processing the TRACE RESULT messages. This value is originally supplied by the initiating node.

Trace Result



RCT = 06 (TRACE RESULT MESSAGE)

TREF = TRACE REFERENCE

Field Length: 2 bytes
Values: Any
Use: This value is included in the TRACE RESULT messages to aid the initiating node in processing the TRACE RESULT messages. This value is originally supplied by the initiating node.

TTCNT = TRACE TRANSIT COUNT

Field Length: 1 byte
Coding: Integer

Values: The TCNT of the TRACE message which caused this TRACE RESULT (0 if caused by a TRACE START).
Use: This aids the receiving node in sorting and collating incoming TRACE RESULT messages.

TSNA = TRACE SOURCE NODE ADDRESS

Field Length: 2 bytes
Values: A node address
Use: This is the address of the source of the TRACE.

TDNA = TRACE DESTINATION NODE ADDRESS

Field Length: 2 bytes
Values: A node address
Use: This is the address of the destination of the TRACE.

MYRF = MY RESISTANCE FACTOR

Field Length: 2 bytes
Coding: Integer
Values: Any
Use: This is the node resistance of the ONA.

REA = REACHABLE

Field Length: 1 byte
Coding: Integer
Values: 00 = UNREACHABLE
        01 = REACHABLE
Use: This value specifies whether the TDNA is reachable from the ONA.

HC = HOP COUNT

Field Length: 1 byte
Coding: Integer
Values: Any
Use: If the TDNA is reachable, this is the hop-count from the ONA to the TDNA over the path on which the TRACE message is forwarded. If the TDNA is UNREACHABLE, the hop-count is MAXHC. If the TDNA is the node sending this message, the hop-count is undefined.

RF = RESISTANCE FACTOR

Field Length: 2 bytes
Coding: Integer
Values: Any
Use: If the TDNA is reachable, this is the RESISTANCE FACTOR from the ONA to the TDNA over the path on which the TRACE message was forwarded. If the TDNA is UNREACHABLE, the resistance factor is MAXRF. If the TDNA is the node sending this message the resistance factor is undefined.

NNA = NEIGHBOR NODE ADDRESS

Field Length: 2 bytes
Values: A node address
Use: If the TDNA is UNREACHABLE or is the ONA, this value is undefined. Otherwise this is the neighbor to whom the TRACE message is forwarded.

LKRF = LINK RESISTANCE FACTOR

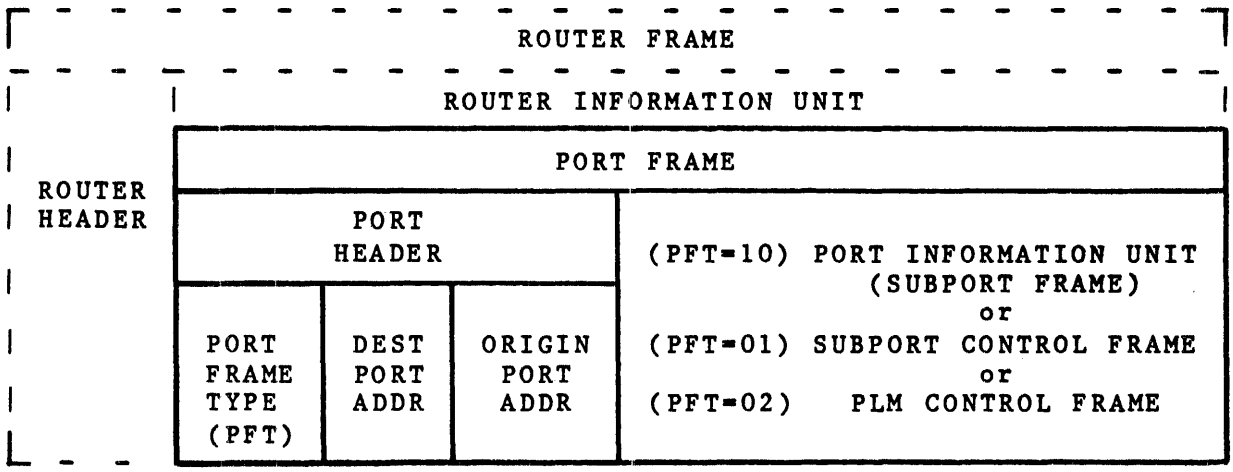
Field Length: 2 bytes
Coding: Integer
Values: Any
Use: If NNA is defined, this is the resistance factor of the link between the ONA and the NNA.

PDN = PDN ID

Field Length: 1 byte
Values: Any
Use: If NNA is defined, this is the PDN ID of the link to the NNA.

PORT FRAME

When the value of the ROUTER FRAME TYPE (RFT) in the ROUTER HEADER is 1, the body of the ROUTER FRAME is a ROUTER INFORMATION UNIT or PORT FRAME.



The PORT FRAME consists of a PORT HEADER and a body. The PORT HEADER consists of these fields: PORT FRAME TYPE, DESTINATION PORT ADDRESS, and ORIGIN PORT ADDRESS. As determined by the PORT FRAME TYPE, the body of the PORT FRAME can be a PORT INFORMATION UNIT (SUBPORT FRAME), a SUBPORT CONTROL FRAME, or a PORT LEVEL MANAGER CONTROL FRAME.

PORT HEADER

Port Frame Type

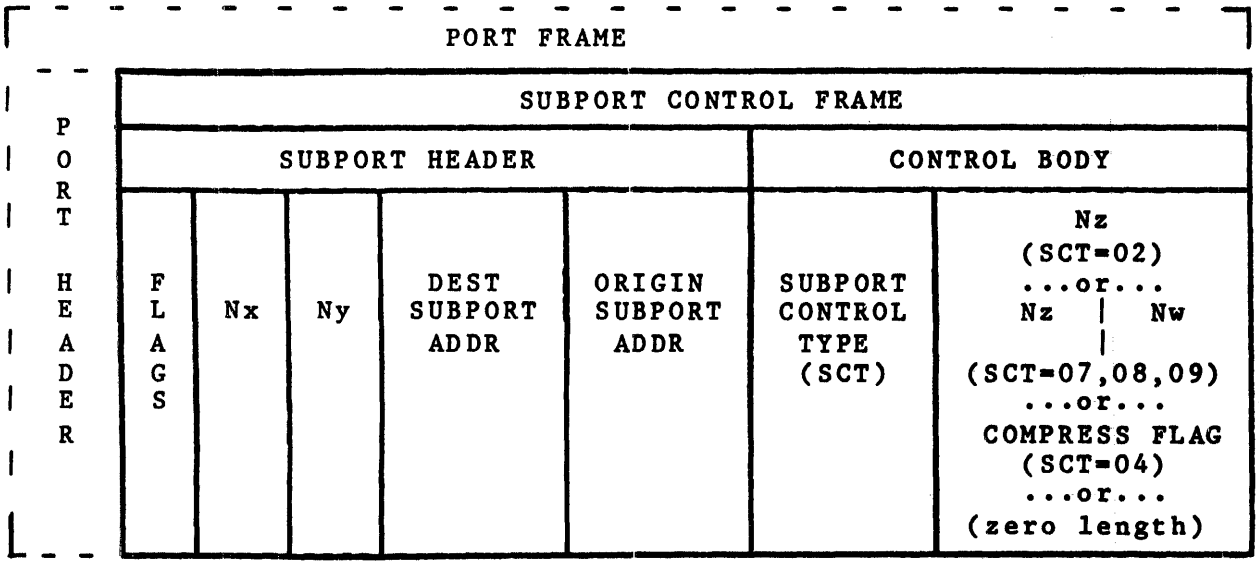
Field Length: 1 byte
Coding: 2 hex digits
Values: 01 = SUBPORT CONTROL FRAME
02 = PORT LEVEL MANAGER CONTROL FRAME
10 = PORT INFORMATION UNIT (SUBPORT FRAME)
All other values are reserved
Use: This field indicates whether the body of the PORT FRAME is a SUBPORT CONTROL FRAME, a PORT LEVEL MANAGER CONTROL FRAME, or a PORT INFORMATION UNIT.

Destination Port Address. Origin Port Address

Field Length: 2 bytes each
Coding: Arbitrary
Values: Each port in a Node must have a unique port address in that Node.
Use: These fields identify a port to the port selector. On outgoing frames, the DESTINATION PORT ADDRESS references the remote port and the ORIGIN PORT ADDRESS references the local port. On incoming frames, the DESTINATION PORT ADDRESS references the local port and the ORIGIN PORT ADDRESS references the remote port. These fields are null (zero) in Port Level Manager Control frames.

SUBPORT CONTROL FRAME

When the value of the PORT FRAME TYPE (PFT) in the PORT HEADER is 02, the body of the PORT FRAME is a SUBPORT CONTROL FRAME. The Subport Control Frame communicates control information between subports.



The SUBPORT CONTROL FRAME consists of a SUBPORT HEADER and a control body. The SUBPORT HEADER consists of these fields: SUBPORT FRAME FLAGS, Nx, Ny, DESTINATION SUBPORT ADDRESS, and ORIGIN SUBPORT ADDRESS. The control body contains a SUBPORT CONTROL TYPE (SCT), and can contain an extension field. Depending on the SUBPORT CONTROL TYPE, the control body extension field can be an Nz field, an Nz field and an Nw field, a COMPRESSION FLAG, or it can be zero length.

Flags

Field Length: 1 byte
Coding: 8 1-bit subfields

Values:

Bit 0 Reserved - Must be 0 (Least Significant bit)

Bit 1 Reserved - Must be 0

Bit 2 TFP = 0 (but ignored)

Bit 3 UNI = 0 (but ignored)

Bit 4 LSI = 0

Bit 5 Reserved - Must be 0

Bit 6 DMI = 1 for SUBPORT CLOSE REQUEST

       = 0 for ABORT

Bit 7 Reserved - Must be 0

Nx

Field Length: 3 Bytes
Coding: Binary numbers
Values: Non-negative integers (0..16777215)
Use: This field is used differently for the various values of SUBPORT CONTROL TYPE.

For SYNC UP 1 and SYNC UP 2:
This field identifies the SEND SEQUENCE NUMBER of the last segment of the sync-area.

For SUBPORT CLOSE REQUEST:
This field contains the SEND SEQUENCE NUMBER for this segment.

For CHANGE COMPRESSION:
This field contains the SEND SEQUENCE NUMBER for this segment. This is used so that the receiving subport can act on the CHANGE COMPRESSION frame in the proper sequence.

For UNNUMBERED ACK:
This field identifies the SEND SEQUENCE NUMBER of the last segment received by the origin subport, that is, the full 24-bit number that would be used for acknowledgment in the Nr field.

For RECEIVE READY, and RECEIVE NOT READY:
This field contains the CONTROL SEQUENCE NUMBER for this segment.

Ny

Field Length: 3 Bytes
Coding: Binary numbers
Values: All zeros
Use: This field is used for close request, abort, and compression change frames.

Destination Subport Address, Origin Subport Address

Field Length: 2 bytes each
Coding: Arbitrary
Values: Each subport in a port must have an address value unique within that port.
Use: This field identifies each subport within a port.

Subport Control Type

Field Length: 1 Byte
Coding: 2 hex digits
Values: 01 = SYNC UP 1
02 = SYNC UP 2
03 = SUBPORT CLOSE REQUEST
04 = CHANGE COMPRESSION
05 = UNNUMBERED ACK
06 = CONTROL ACK
07 = RECEIVE READY
08 = RECEIVE NOT READY
09 = SUBPORT ABORT
Use: This field indicates the use for the SUBPORT CONTROL FRAME.

Nz

Field Length: 3 Bytes
Coding: Binary numbers
Values: Non-negative integers (0..16777215)
Use: This field is used differently for the various values of SUBPORT CONTROL TYPE.

For SYNC UP 2:
This field identifies the SEND SEQUENCE NUMBER of the first segment of the sync-area.

For RECEIVE READY, RECEIVE NOT READY, and CONTROL ACK:
This field identifies the CONTROL SEQUENCE NUMBER of the highest numbered RECEIVE READY or RECEIVE NOT READY frame received by the origin subport.

Nw

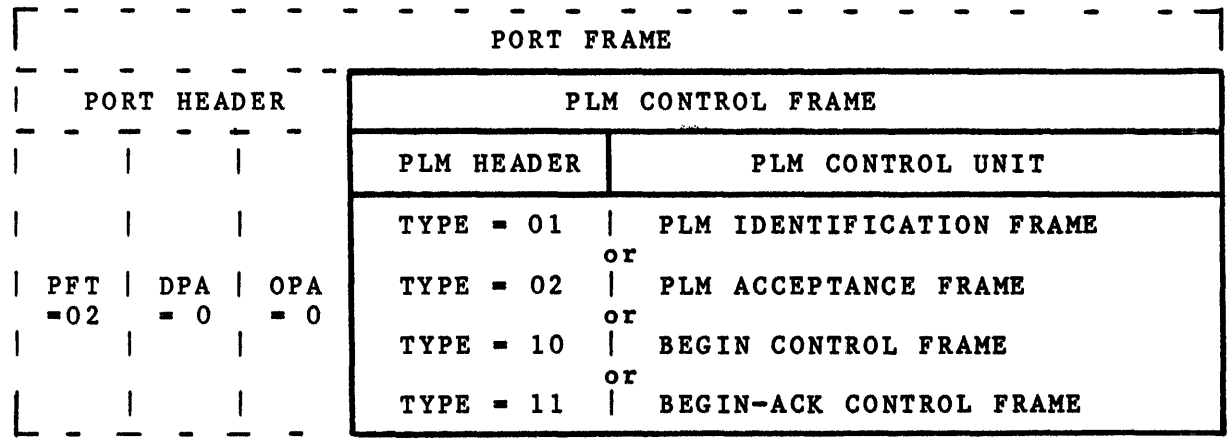
Field Length: 3 bytes
Coding: Binary numbers
Values: Non-negative integers (0..16777215)
Use: For RECEIVE READY, RECEIVE NOT READY, and CONTROL ACK: This field identifies the CONTROL SEQUENCE NUMBER of the highest numbered RECEIVE READY or RECEIVE NOT READY frame received by the origin subport.

Compression Flag

Field Length: 1 byte
Coding: 2 hex digits
Values: 00 = NO COMPRESSION
01 = COMPRESS DATA
Use: This field is used for CHANGE COMPRESSION control frames. The value indicates whether or not data compression is to be used in subsequent information frames received.

PLM CONTROL FRAME

When the value of the PORT FRAME TYPE (PFT) in the PORT HEADER is 02, the body of the PORT FRAME is a PORT LEVEL MANAGER (PLM) CONTROL FRAME. The PLM Control Frame is used to communicate control information between PORT LEVEL MANAGERS.



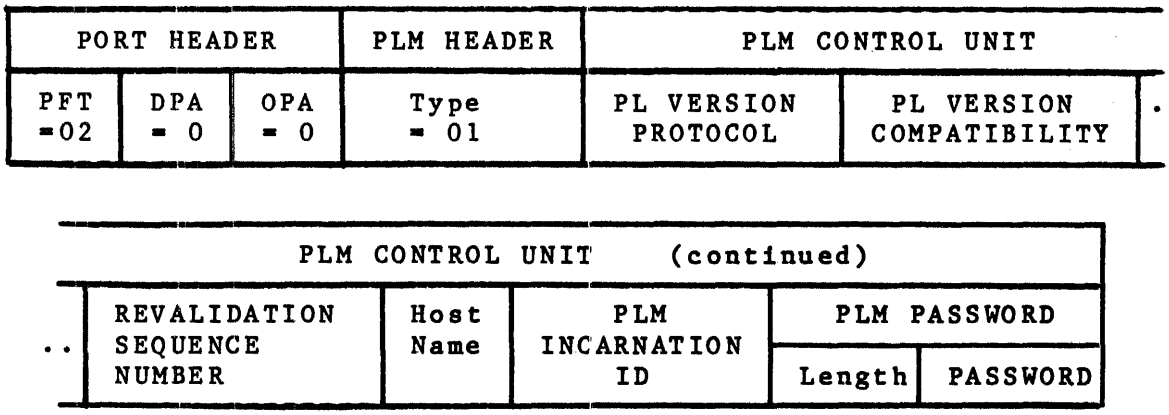
The value in the PLM HEADER identifies the PLM CONTROL UNIT as a PLM IDENTIFICATION FRAME (01), a PLM ACCEPTANCE FRAME (02), a BEGIN CONTROL FRAME (10), or a BEGIN-ACK CONTROL FRAME (11). The values of the DESTINATION PORT ADDRESS (DPA) and the ORIGIN PORT ADDRESS (OPA) are both 0, since no ports are involved.

There are two classes of control frames used by the PORT LEVEL MANAGER. The first consists of GREETING CONTROL frames, used during PORT LEVEL MANAGER to PORT LEVEL MANAGER validation. The second class contains BEGIN and BEGIN-ACK control frames and establishes a PLM-to-PLM subport dialog.

Greeting Control Frames

There are two Greeting Control Frames, PLM IDENTIFICATION and PLM ACCEPTANCE. In both frames of a matching (PLM IDENTIFICATION, PLM ACCEPTANCE) pair, most of the data in the control unit is defined by the PORT LEVEL MANAGER that generates the PLM IDENTIFICATION frame.

PLM IDENTIFICATION FRAME



PLM HEADER

Field Length: 1 Byte
Coding: 2 hex digits
Values: 01 = PLM IDENTIFICATION
Use: Identifies the PLM control frame as a PLM IDENTIFICATION frame.

PL VERSION PROTOCOL

Field Length: 1 Byte
Coding: Integer
Values: 1..255
Use: The version-protocol level used for communication with the remote PORT LEVEL MANAGER during the current exchange of greetings.

PL VERSION COMPATIBILITY

Field Length: 3 Bytes
Coding: 24 Binary bit mask

Values: The high order bit of the mask corresponds to the protocol level "(PL VERSION PROTOCOL)-1", and the low order bit corresponds to "(PL VERSION PROTOCOL)-24". A bit being on (TRUE) means that the corresponding protocol level is supported by this software.
Use: Identifies the BNA standard protocol levels (below PL VERSION PROTOCOL) at which the PORT LEVEL software is capable of communicating with the remote PORT LEVEL MANAGER during the current exchange of greetings.

REVALIDATION SEQUENCE NUMBER

Field Length: 1 Byte
Coding: Integer
Values: 1..255
Use: Controls requests for revalidation due to previous loss of communications.

Host Name

Field Length: 17 Bytes
Coding: Non-blank graphic characters, left-justified and blank filled.
Values: Name
Use: Identifies the Host Name of the Host on behalf of which the PLM IDENTIFICATION Frame is sent.

PLM INCARNATION ID

Field Length: 8 Bytes
Coding: 64 Binary bit string
Values: Any, except "all bits off"
Use: Identifies the incarnation of the PORT LEVEL MANAGER that is currently running on the local Host. The "incarnation" changes whenever the PLM PHASE changes from a value of ISOLATED to any other value or when the PORT LEVEL MANAGER starts execution.

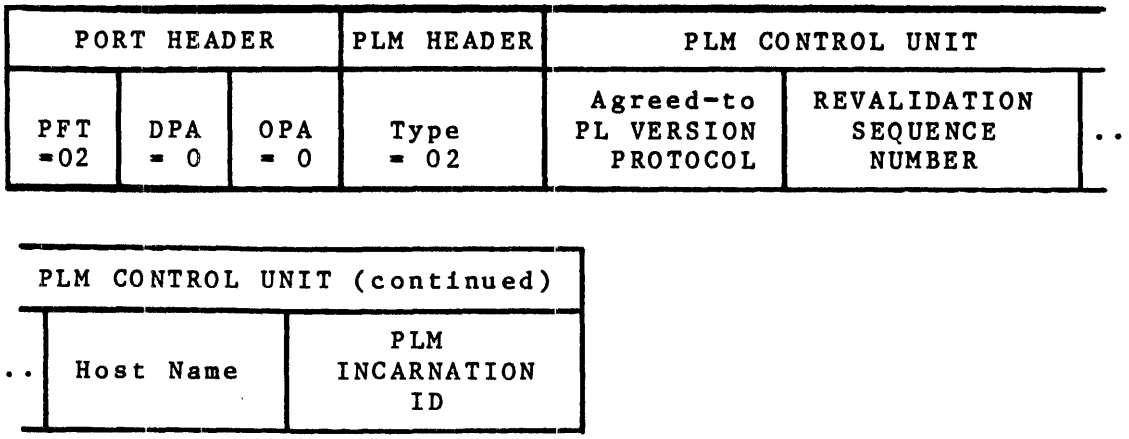
PLM PASSWORD LENGTH

Field Length: 1 byte.
Coding: Integer
Values: 0...17
Use: Identifies the length of the PLM PASSWORD field.

PLM PASSWORD

Field Length: Variable, maximum 17 Bytes.
Coding: EBCDIC string
Values: Any EBCDIC characters
Use: Identifies the password that the Host receiving the PLM IDENTIFICATION Frame uses to validate the Host that (allegedly) sent that frame.

PLM ACCEPTANCE FRAME



PLM HEADER

Field Length: 1 Byte
Coding: 2 hex digits
Values: 02 = PLM ACCEPTANCE
Use: Identifies the PLM CONTROL Frame as a PLM ACCEPTANCE Frame.

Agreed-to PL VERSION PROTOCOL

Field Length: 1 Byte
Coding: Integer
Values: 1..255
Use: The version-protocol level agreed to be used for communication with the remote PORT LEVEL MANAGER during the current exchange of greetings.

REVALIDATION SEQUENCE NUMBER

Field Length: 1 Byte
Coding: Integer
Values: 1..255
Use: Identifies the value of the REVALIDATION SEQUENCE NUMBER in the frame being accepted.

Host Name

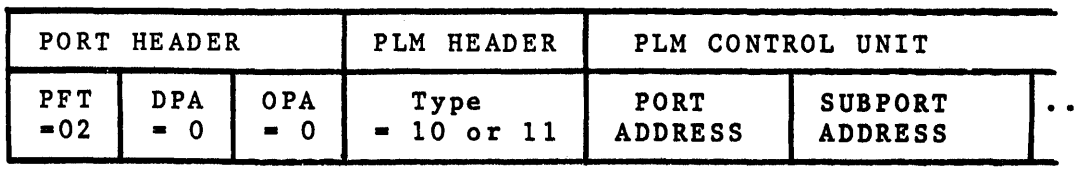
Field Length: 17 Bytes
Coding: Non-blank graphic characters, left-justified and blank filled.
Values: Name
Use: The name of the Host sending the PLM IDENTIFICATION Frame being accepted.

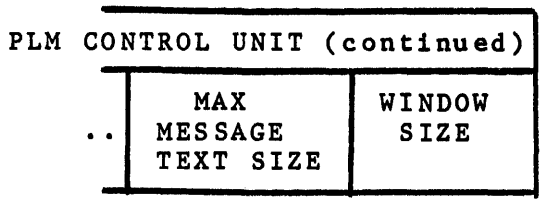
PLM INCARNATION ID

Field Length: 8 Bytes
Coding: 64 Binary bit string
Values: Any, except "all bits off"
Use: Identifies the incarnation of the PORT LEVEL MANAGER that is currently running on the Host that sent the PLM IDENTIFICATION Frame being accepted.

BEGIN/BEGIN-ACK Control Frame

The BEGIN control frame is used to initiate the establishment of a PLM-to-PLM subport dialog. The BEGIN-ACK control frame is the response to a BEGIN control frame. The BEGIN control frame and the BEGIN-ACK control frame are identical in format except for the PLM HEADER Type. In both frames, the ADDRESSES used in the PLM CONTROL UNIT are the ADDRESSES of the PLM's subport being opened at the Node that originated the BEGIN Control Frame (of a matching [BEGIN, BEGIN-ACK] pair).





PLM HEADER

Field Length: 1 Byte
Coding: 2 hex digits
Values: 10 = BEGIN
11 = BEGIN-ACK
Use: Identifies the PLM Control Frame as a BEGIN or BEGIN-ACK Control Frame.

PORT ADDRESS and SUBPORT ADDRESS

Field Length: 2 Bytes
Coding: Integer
Values: 1..65534
Use: Addresses at the origin NODE of the BEGIN Control Frame for PLM - PLM communication.

MAX MESSAGE TEXT SIZE

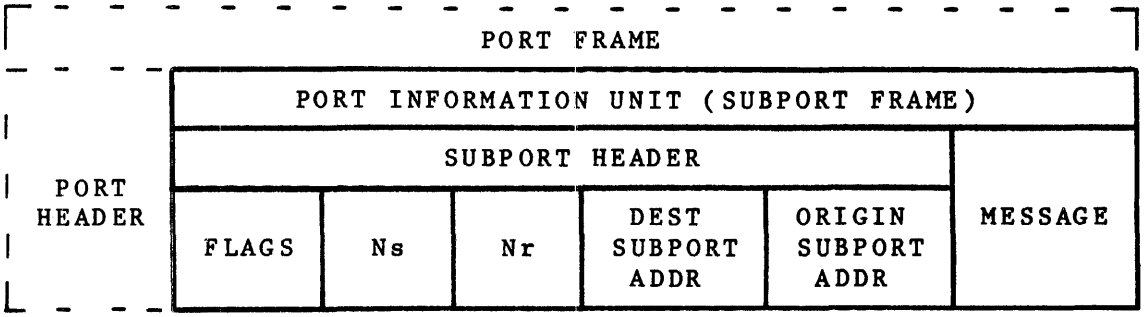
Field Length: 2 bytes
Coding: Integer
Values: 0..65535
Use: This value in the BEGIN frame is the desired value. In the BEGIN-ACK frame it is the agreed-upon value, the minimum of the local and remote desired values.

WINDOW SIZE

Field Length: 1 byte
Coding: Integer
Values: 1..60
Use: This value in the BEGIN frame is the desired value. In the BEGIN-ACK frame, it is the agreed-upon value, the minimum of the local and the remote desired values.

PORT INFORMATION UNIT (SUBPORT FRAME)

When the value of the PORT FRAME TYPE (PFT) in the PORT HEADER is 10, the body of the PORT FRAME is a PORT INFORMATION UNIT (SUBPORT FRAME).



The PORT INFORMATION UNIT (SUBPORT FRAME) consists of a SUBPORT HEADER and a body. The SUBPORT HEADER consists of these fields: SUBPORT FRAME FLAGS, SEND SEQUENCE NUMBER, RECEIVE SEQUENCE NUMBER, DESTINATION SUBPORT ADDRESS, and ORIGIN SUBPORT ADDRESS. The body of the SUBPORT FRAME can be a segment of the Host Services message or User message, a segment of a PLM message, or it can be empty. If the message is short enough, it is the entire message.

The type of message in the SUBPORT FRAME body is determined by the PORT ADDRESS in the PORT HEADER. If the SUBPORT is a PORT LEVEL MANAGER (PLM) subport, the message is a PLM message. Otherwise, it is a message for a Host Services subport or User subport.

Subport Frame Flags

Field Length: 1 byte
Coding: 8 1-bit subfields

Use:

Bit 0 Reserved - Must be 0 (Least Significant bit)

Bit 1 Reserved - Must be 0

Bit 2 (TFP = TEXT FRAME SEGMENT PRESENT)
If TRUE (=1), the body of the SUBPORT FRAME is present. If FALSE (=0), it is zero length.

Bit 3 (UNI = USE SEND AND RECEIVE SEQUENCE NUMBERS)
If TRUE (=1), the SEND and RECEIVE SEQUENCE NUMBER fields contain SEQUENCE NUMBERS for acknowledging and ordering the frames; if FALSE (=0), these fields can be used for some other purpose.

Bit 4 (LSI = LAST SEGMENT INDICATOR)
If a TEXT FRAME is present (see bit 2), this field flags the last segment of the TEXT FRAME. If TRUE (=1), the SUBPORT FRAME contains the last segment of the segmentable TEXT FRAME. If FALSE (=0), it does not. If a TEXT FRAME is not present, this subfield is set FALSE by the subport when generating the frame, and is not interpreted when receiving the frame.

Bit 5 Reserved - Must be 0

Bit 6 (DMI = DEMAND ACKNOWLEDGMENT)
If TRUE (=1), the sending subport wants to receive an acknowledgment as soon as possible from the remote subport. If FALSE (=0), acknowledgment can follow in the normal manner.

Bit 7 Reserved - Must be 0

Send Sequence Number (NS or Ns)

Field Length: 3 bytes
Coding: Binary number
Values: Non-negative integers (0..16777215)
Use: The local subport identifies outgoing SUBPORT FRAMES using consecutive SEND SEQUENCE NUMBERS (modulo 16777216). The corresponding remote subport uses this number for FIFO reordering and for acknowledging receipt of SUBPORT FRAMES.

Receive Sequence Number (NR or Nr)

Field Length: 1 byte
Coding: Binary number
Values: Non-negative integers (0..255)
Use: The local subport places a number in the RECEIVE SEQUENCE NUMBER field in the outgoing frames. This number corresponds to the SEND SEQUENCE NUMBER (modulo 256) in the last frame received in a series of frames in which there is no missing frame. This number is updated each time a frame is retransmitted.

Destination Subport Address. Origin Subport Address

Field Length: 2 bytes each
Coding: Arbitrary
Values: each subport in a port must have an address value unique within that port.
Use: This field identifies each subport to the Subport Demultiplexor.

PLM Messages

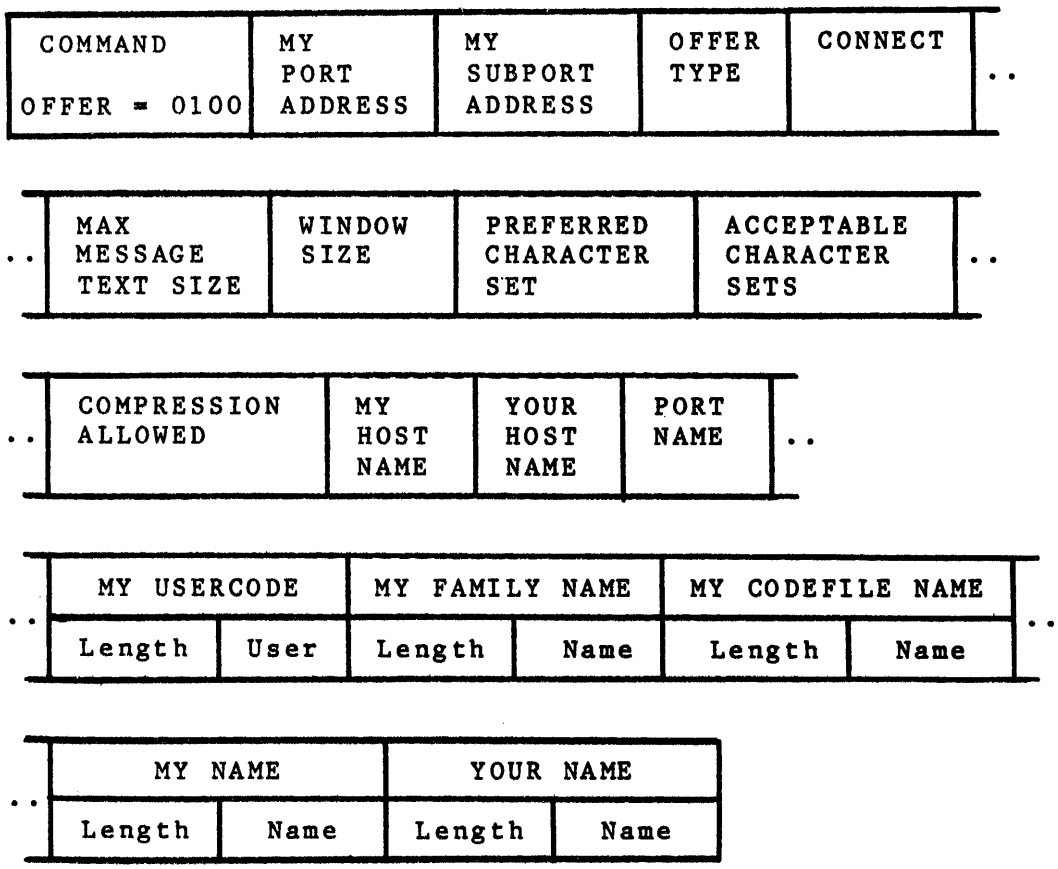
The information frames used in communications between Port Level Managers appear as normal information frames. The PORT INFORMATION UNITs in these frames contain message text that is described below.

These messages are exchanged using the PLM's subports, which were opened by an exchange of BEGIN and BEGIN-ACK control frames.

The possible values of the COMMAND field (in hex) are:

OFFER	= 0100
RESCIND OFFER	= 0101
MATCH FOUND	= 0110
NO MATCH	= 0111
ACCEPT MATCH	= 0120
REFUSE MATCH	= 0121
DEACTIVATE SUBPORT	= 0131
TERMINATE PLM DIALOG	= 01FF

OFFER MESSAGE



MY ... ADDRESSES

Field Length: 2 bytes
Coding: Integer
Values: The addresses of the port and subport being offered for communication.

OFFER TYPE

Field Length: 1 byte
Coding: 2 hex digits
Values: 01 = BE PATIENT
02 = JUDGE QUICKLY
Use: Indicates to the receiving PLM whether the sending PLM requires an immediate answer.

CONNECT

Field Length: 1 byte
Coding: 2 hex digits
Value: Undefined

PREFERRED CHARACTER SET (in hex):

Field Length: 1 byte
Coding: 2 hex digits
Values: 00 = DON'T CARE
01 = EBCDIC
02 = ASCII
Use: Identifies the character set that the message originator prefers to use for communications.

ACCEPTABLE CHARACTER SETS:

Field Length: 1 byte
Coding: 8 binary bits
Values: Character set "n" is acceptable if bit "n" is 1, where the bits are numbered 1 through 8, starting with the rightmost (least-significant) bit. The value of "n" for a given character set is as specified by the values for PREFERRED CHARACTER SET. All bits for "undefined" character set numbers must be off (zero)
Use: Identifies the character set that the message originator will accept for communications. This value must include the character set identified in the PREFERRED CHARACTER SET.

COMPRESSION ALLOWED:

Field Length: 1 byte
Coding: 8 binary bits
Values: This field consists of eight (8) bits, numbered 0 (least significant) through 7 (most significant). The meaning of these bits is as follows:

BIT 0: TRUE (=1) if sending Host permits compression of runs of identical characters
FALSE (=0) if such compression is not permitted

BITS 1-7: Reserved (must be 0)
Use: Indicates whether the HOST originating the message supports data compression/expansion.

MY HOST NAME, YOUR HOST NAME, PORT NAME:

Field Length: 17 bytes
Coding: Non-blank EBCDIC graphic characters, left-justified in a blank-padded field 17 characters long.

MY USERCODE, MY FAMILY NAME:

Field Length: VARIABLE, maximum 17 bytes
Coding: EBCDIC string

MY CODEFILE NAME, MY NAME, YOUR NAME:

Field Length: VARIABLE, maximum 255 bytes
Coding: EBCDIC string

RESCIND OFFER MESSAGE

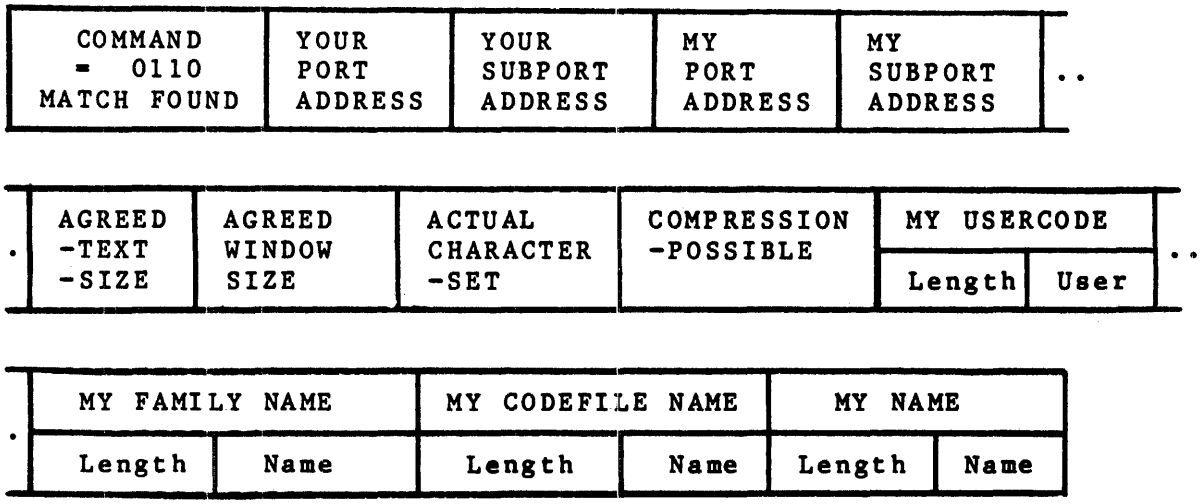
MY ... ADDRESSES

Field Length: 2 bytes
Coding: Integer
Values: Same as in the corresponding OFFER message.

RESCIND REASON

Field Length: 1 byte
Coding: 2 hex digits
Values: 01 = SUBPORT CLOSED
02 = GOING AWAY
Use: Identifies the reason for the RESCIND OFFER Command.

MATCH FOUND MESSAGE



This message requires an answer, which can be ACCEPT MATCH or REFUSE MATCH.

YOUR ... ADDRESSES

Field Length: 2 bytes
Coding: Integer
Values: The addresses of the matched remote port and subport.

MY ... ADDRESSES

Field Length: 2 bytes
Coding: Integer
Values: The addresses of the port and subport
local to the matching PORT LEVEL MANAGER.

AGREED TEXT SIZE

Field Length: 2 bytes
Coding: Integer
Values: 1..65535
Use: The result of negotiation.

AGREED WINDOW SIZE

Field Length: 1 byte
Coding: Integer
Values: 1..60
Use: The result of negotiation.

ACTUAL CHARACTER SET

Field Length: 1 byte
Coding: 2 hex digits
Values: 00 = DON'T CARE
        01 = EBCDIC
        02 = ASCII
Use: The result of negotiation, indicating the agreed-upon character set for the subport dialog.

COMPRESSION POSSIBLE

Field Length: 1 byte
Coding: 8 binary bits
Values:

Bit 0: (Least significant bit)
        TRUE (=1) if compression of runs of identical characters is permitted,
        FALSE (=0) if such compression is not permitted.

Bits 1-7:
        Reserved - must be 0.

Use: The result of negotiation, indicates whether or not data compression/expansion is supported.

MY USERCODE, MY FAMILY NAME:

Field Length: Variable, maximum 17 bytes
Coding: EBCDIC string

MY CODEFILE NAME, MY NAME:

Field Length: Variable, maximum 255 bytes
Coding: EBCDIC string

NO MATCH MESSAGE

COMMAND = 0111 NO MATCH	YOUR PORT ADDRESS	YOUR SUBPORT ADDRESS	NO MATCH REASON

YOUR ... ADDRESSES

Field Length: 2 bytes
Coding: Integer
Values: The addresses of the port and subport which were OFFERed.

NO MATCH REASON (in hex):

Field Length: 1 byte
Coding: 2 hex digits
Values:01 = NO MATCH FOUND
       02 = GOING AWAY

ACCEPT MATCH MESSAGE

COMMAND = 0120 ACCEPT MATCH	MY PORT ADDRESS	MY SUBPORT ADDRESS	YOUR PORT ADDRESS	YOUR SUBPORT ADDRESS

The ADDRESSES in this message are the same addresses, in the same order, as the addresses in the MATCH FOUND message to which this is an answer; however, the MY/YOUR sense has reversed.

MY ... ADDRESSES

Field Length: 2 bytes
Coding: Integer

YOUR ... ADDRESSES

Field Length: 2 bytes
Coding: Integer

REFUSE MATCH MESSAGE

COMMAND = 0121 REFUSE MATCH	MY PORT ADDRESS	MY SUBPORT ADDRESS	YOUR PORT ADDRESS	YOUR SUBPORT ADDRESS	REFUSAL REASON

The ADDRESSES in this message are the same addresses, in the same order, as the ones in the MATCH FOUND message to which this message is an answer; however, the MY/YOUR sense has reversed.

YOUR ... ADDRESSES

Field Length: 2 bytes
Coding: Integer

MY ... ADDRESSES

Field Length: 2 bytes
Coding: Integer

REFUSAL REASON

Field Length: 1 byte
Coding: 2 hex digits
Values: 01 = INACCESSIBLE (failed ACCESSIBILITY check)
        02 = CAN'T FIND IT (not on CANDIDATES FOR MATCH list)
        03 = DISAGREEMENT (bad negotiated attributes)

DEACTIVATE SUBPORT MESSAGE

COMMAND = 0131 DEACTIVATE SUBPORT	MY PORT ADDRESS	MY SUBPORT ADDRESS	YOUR PORT ADDRESS	YOUR SUBPORT ADDRESS	DEACTI VATION REASON

DEACTIVATION REASON

Field Length: 1 byte
Coding: 2 hex digits
Values: 01 = SUBPORT CLOSED
        02 = GOING AWAY

TERMINATE PLM DIALOG MESSAGE

COMMAND = 01FF TERMINATE PLM DIALOG	FLAGS	TERMINATION REASON

FLAGS

Field Length: 1 byte
Coding: 8 Binary bits
Values:

Bit 0 (Least Significant):
        TRUE (=1) if this is a demand for dialog termination
       FALSE (=0) if this is a request for termination

Bits 1-7:
        Reserved - must be 0.

TERMINATION REASON

Field Length: 1 byte
Coding: 2 hex digits
Values: 01 = INACTIVITY
        02 = ISOLATING MYSELF
        03 = BREAKING OUR CONNECTION

INDEX

INDEX (CONT)

INDEX (CONT)

INDEX (CONT)

INDEX (CONT)

INDEX (CONT)

Documentation Evaluation Form

Title: Burroughs Network Architecture (BNA) Architectural Description Reference Manual (Vol 1)

Form No: 1132172

Date: April 1981

Burroughs Corporation is interested in receiving your comments and suggestions regarding this manual. Comments will be utilized in ensuing revisions to improve this manual.

Please check type of Suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments:

From:

Name _____

Title _____

Company _____

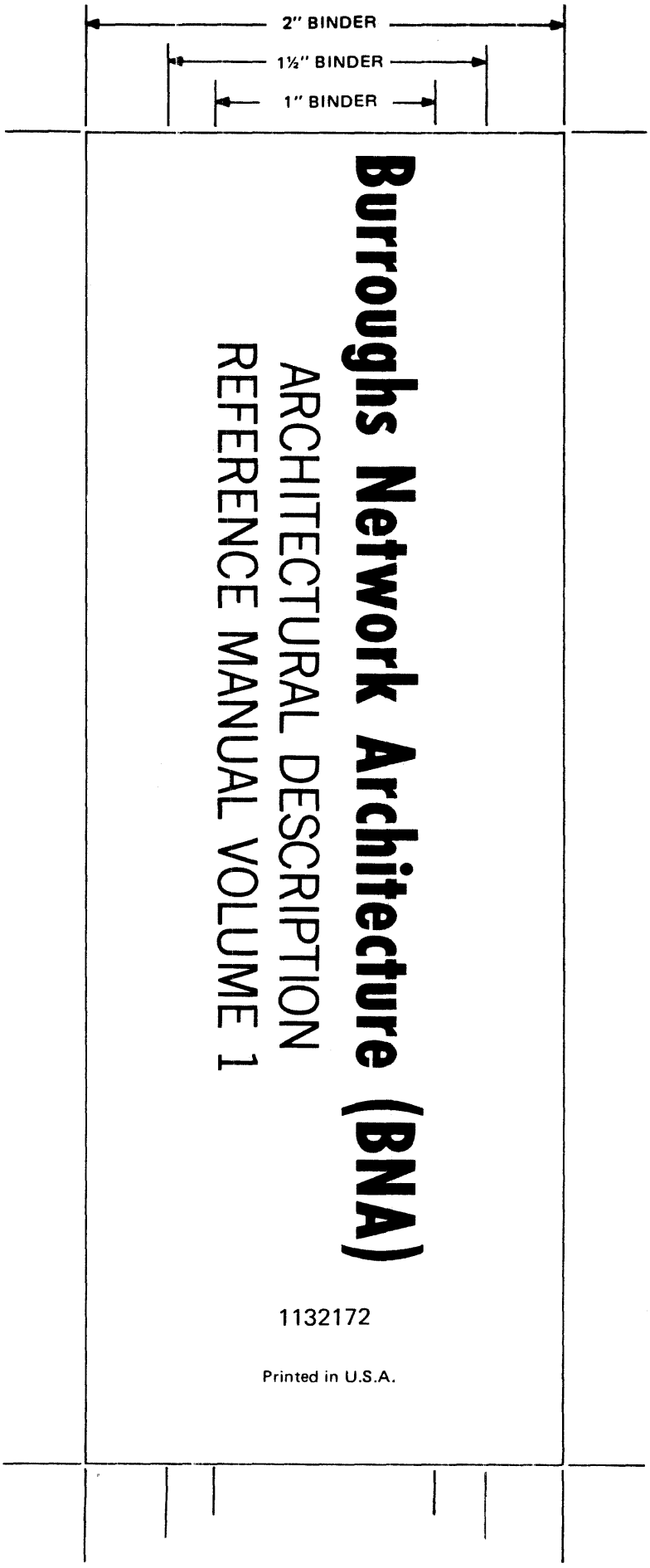
Address _____

_____

Phone Number _____ Date _____

Remove form and mail to:

Documentation Dept, TIO - East
Burroughs Corporation
Box CB7
Malvern, PA 19355


2" BINDER
1½" BINDER
1" BINDER
Burroughs Network Architecture (BNA)
ARCHITECTURAL DESCRIPTION
REFERENCE MANUAL VOLUME 1
1132172
Printed in U.S.A.