DECnet™ DIGITAL Network Architecture (Phase V)

General Description

Order No. EK-DNAPV-GD

September 1987

This document describes the design of the DIGITAL Network Architecture that serves as a model for Phase V DECnet implementations. It includes descriptions of functions, protocol messages, and operations.

digital™

To order additional copies of this document, contact your local Digital Equipment Corporation Sales Office

digital equipment corporation • maynard, massachusetts

Contents

Preface

This book is an overview of the DIGITAL Network Architecture (DNA). DNA is a model of structure and functions upon which DECnet implementations are based. DECnet is a family of communications software and hardware products that enable DIGITAL operating systems and computers to function in a network with other DIGITAL systems and with systems manufactured by other vendors.

A DECnet network is a group of computer systems with associated operating systems, DECnet software, and communication hardware that are connected to each other by physical channels, or *lines*. Each computer in a network containing a DECnet implementation is called a *system*. A *network* therefore consists of connected systems and lines. DNA defines standard protocols, interfaces, and functions that enable DECnet systems to share data and access others' resources, programs, and functions.

The DNA Model incorporates both a set of proprietary protocols and interfaces, and a set of protocols and services defined for Open System Interconnection by the International Standards Organization.

This book describes Phase V of DNA. Phase V provides compatibility with the previous version of DNA, Phase IV.

DNA supports a broad range of applications and a variety of network topologies. (A network *topology* is a particular configuration of systems and lines.) User documentation and marketing brochures describe in detail various types of application and specific programming and network management information.

This document summarizes the design and structure of DNA and serves as an introduction to the DNA functional specifications. It is intended for readers with a knowledge of communications technology who desire an understanding of the overall DNA structure. A glossary at the end of the document defines many DNA terms. Additionally, many DNA terms are italicized and explained in the text at their first occurrence.

Phase V of DNA defines a rich set of functions. Some implementations will not need all of these functions. There are also some aspects of Phase V which may not be present in initial implementations.

Chapter 1

Introduction

The Digital Network Architecture (DNA) defines the means by which computer systems can communicate with each other to form a distributed processing system. It specifies:

- the communication protocols by which systems exchange information, including the rules for constructing and interpreting messages.
- the internal interfaces which allow one protocol to be described and built in terms of the services provided by another.
- the policies and mechanisms by which systems adapt to changing loads and configurations to make the best use of available resources.
- the means by which a distributed network can be managed, both locally and remotely.

This General Description relates to *Phase V* of the DNA model. Phase V is an evolutionary step from the previous version, Phase IV. There are two major new technical directions in Phase V: integration of the OSI standards, and support for increasing network sizes.

OSI ("Open Systems Interconnection") is a set of international standards developed by the International Standards Organization (ISO). Its goal is to allow different vendors' computer systems to be freely interconnected. The standards for the lower layers, providing the basic communications functions, are now substantially complete. DNA uses these OSI standards wherever possible so that Digital computer systems can be used in multi-vendor networks and distributed processing systems.

The increasing importance of networking means that computer networks are continually getting larger. Where a few years ago networks of even a dozen systems were rare, today networks of hundreds of systems are commonplace and several much larger networks, up to tens of thousands of systems, are in use. Furthermore, this increase in the size and number of networks means that they can no longer be operated exclusively by specialists as they have in the past. DNA includes many capabilities to simplify the design, construction and operation of large networks. Of particular importance is the Naming Service, which automates the distribution of critical shared information such as the network addresses of resources, and their protocol capabilities.

1.1 Necessity for an Architecture

A system as complex as a DECnet network could not be built without an architecture. The architecture is necessary for several reasons:

- There is a wide variety of computer hardware, communication devices and operating systems. These are built by separate implementation groups. Communication is only possible if they use a common set of specifications.
- Industry Standards (in particular, OSI) are a vital part of computer networking. The architecture specifies how these standards are used and permits individual standards to be introduced and integrated without disrupting the overall function of the network.
- Communications technology is constantly developing. A network product family needs to be able to evolve to include these new technologies. The architecture makes this possible by imposing a modular structure whereby changes in one area do not affect other areas.
- The modular structure of the architecture permits high-performance functions to be implemented in hardware, in a system-independent way. For example, Local Area Network adapters implement the LAN data link protocols.

1.2 Design Goals

DNA was designed to support the following goals.

Conceal network operation from the user. The operation of a large network is necessarily complex, but to the user it should appear simple. It then becomes very natural to build distributed applications.

Support a wide range of applications. Almost any application may be distributed across a network. DNA supports a great variety of applications, and allows new ones to be developed freely.

Support a wide range of communication facilities. The technology of data communications is constantly evolving. Not only are greater speeds becoming available, but the nature of the services is also changing. This is clearly illustrated by the development of Local Area Networks and X.25 packet-switched networks. The structure of DNA allows these new technologies to be integrated and supported without major change to the remainder of the architecture. In Phase V, DNA supports traditional synchronous data links, Local Area Networks, X.25 packet-switched networks and X.21 circuit-switched networks.

Support a wide range of network topologies. DNA does not impose any particular topology (such as star, mesh or hierarchical). It makes the best use of a topology constructed to meet the performance and cost requirements of each network.

Make maximum use of standards. Standards are critical to successful computer networks. Wherever feasible, DNA uses OSI standards rather than proprietary means to attain its goals. Where OSI standards are being worked on but are not yet approved, DNA is designed to make future migration as straightforward as possible.

Require minimum management intervention. Wherever possible, DNA allows a network to operate without operator or network management intervention. This is particularly important for user-oriented systems, such as personal workstations. The architecture includes algorithms and mechanisms for automatic determination of operational characteristics. Examples include the dynamic routing algorithm, and the automatic determination of a system's address.

Be manageable. Some functions cannot be automated, and for those which can it is sometimes necessary to override the automatic choice. This is particularly true for large or complex networks. DNA permits remote management of all network functions. The architecture does not impose any particular style of management (such as centralized or hierarchical), but leaves this to the user's choice.

Permit growth. DNA allows a network to grow from just two systems to a huge global network without disruption or major reconfiguration. To the systems attached to the network, such growth is completely transparent.

Permit migration. As new versions of the architecture and its implementations are developed, existing networks must migrate to them. Migration of a large network may take months or even years, and during this period it must continue to operate normally. Each Phase of DNA will fully interwork with the previous Phase. All Phase V systems can communicate with the Phase IV systems in a network throughout its migration period.

Be subsettable. Not all systems need to implement all functions of the architecture. DNA allows an implementation to omit certain functions, particularly those which are expensive to provide and which do not need to be in every system for the network to function.

Be extensible. DNA allows for the incorporation of future technology changes in hardware, software and standards.

Be highly available. DECnet networks can be configured to maintain operation even if some lines or systems fail, with minimum prior planning or operator intervention.

Be highly distributed. The major functions of DNA, such as routing and network management, are not centralized in a single system in the network. This in turn increases the availability of the network.

Allow for security. DNA allows for security at several levels. User access control is implemented in most DECnet products.

1.3 The Structure of DNA

1.3.1 Layered Architecture

DNA is a *layered* architecture. This means that the functions required to achieve its goals are divided into related and logically coherent groups called *layers*. The layers are built on top of one another, so that one layer makes use of services provided by the one beneath it. The detailed protocols and other mechanisms of a layer are hidden from the layers above and below – all that is seen by the layer above is the *service* it provides. This gives the property of *layer independence,* so that very substantial changes can be made to one layer (such as changing the protocol used) without affecting other layers. A layered architecture is much better able to meet the goals set for DNA than one in which functions are performed in a less structured manner.

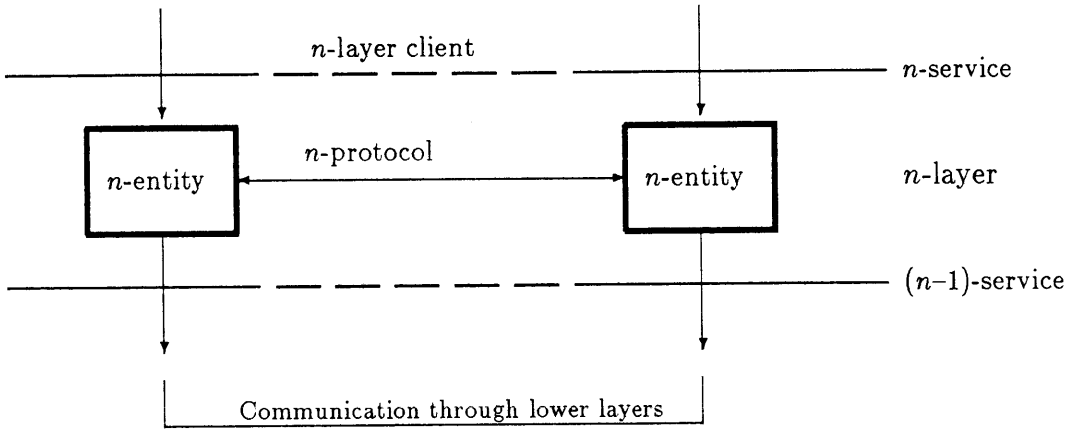


Figure 1.1: Principles of a single layer

Figure 1.1 illustrates the principles of a layer, using the terminology of the OSI Reference Model. It applies to any layer – for generality, the term n-layer is used to refer to an unspecified layer of the architecture. Part of the layer exists in each of the communicating systems. Within one system, the component responsible for the n-layer is called the *n-entity*. The two n-entities communicate with each other using a protocol, called the *n-protocol.* This protocol is conveyed using the services of the next lower layer. All layers provide a data transfer service, although the details vary. Some layers provide additional services such as connection management. In general, the n-protocol is carried as data by the $(n-1)$-service. The lower layer has no knowledge of, or interest in, the content of the data it conveys. Each n-entity provides the *n-service* to a local *client.* The client may be either the $(n+1)$-layer, or an end-user.

When a layer carries data on behalf of a client, it also adds a header containing *Protocol Control Information* (PCI). The PCI contains information which the n-entities need to control each others' operation. It typically includes message sequence numbers and information about the precise service requested. The combination of the PCI and the user data

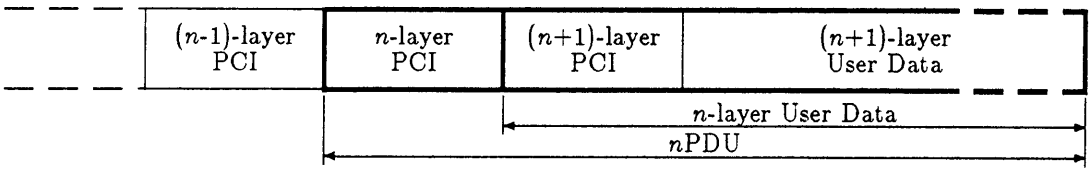


Figure 1.2: Nesting of Protocol Control Information

is called a *Protocol Data Unit* (PDU) – a PDU of layer n is called an nPDU. Thus a PDU of the Transport layer, for example, is called a TPDU. Often, other names are used for PDUs at a specific layer, such as *packet* or *frame*. Since there are several layers involved in providing a service to the user, the data actually sent across a physical channel includes the PCI for several layers, each carried as data by the lower layer. This is illustrated in figure 1.2. In the Data Link layer, some PCI is also carried in a *trailer* which follows the user data.

The data passed to a layer by its client is called a *Service Data Unit* (SDU). SDUs pass *within* a system, between a layer and its client. They are not visible from outside the system. In contrast, PDUs pass *between* systems, between the communicating entities of a layer. They are visible outside the system as the data flowing "across the wire" between the two communicating systems. Section 1.3.3 describes this in terms of the layers of DNA. In the simplest case, a PDU consists simply of the corresponding SDU together with a header containing PCI. Some layers have the ability to divide an SDU into smaller units, sending each in its own PDU. For example, this allows a layer to convey very large SDUs even though the layer beneath it has a relatively small maximum SDU size.

In DNA, a layer may use more than one protocol, for reasons of compatibility with earlier versions or to accommodate alternative option selections from standards. In addition, in some layers one protocol may be a client of another *within* the layer. A *module* is an entity which implements a specific protocol, whether or not it is the only protocol in its layer. Each protocol in DNA is defined as a module. A module provides a service to its clients, even if they may be further modules within the same layer.

Layers and modules generally have parameters which can be selected by a client, for example at the time a connection is made. Often, a client needs a particular service but does not need to be concerned with the details of individual parameters. DNA provides *templates* in many modules, which are named collections of parameters and their values defined using network management. A client can reference a template by its name, gaining the same effect as if it had specified the parameter values individually. This gives a further degree of layer independence, since the architecture for one module need not reflect in detail the parameters of a supporting module. For example, X.25 provides many options to do with the exact way the X.25 connection is made. Users of X.25, such as the Network

Routing module, can access these without having to reflect the options explicitly in their own definition.

1.3.2 The Layers of DNA

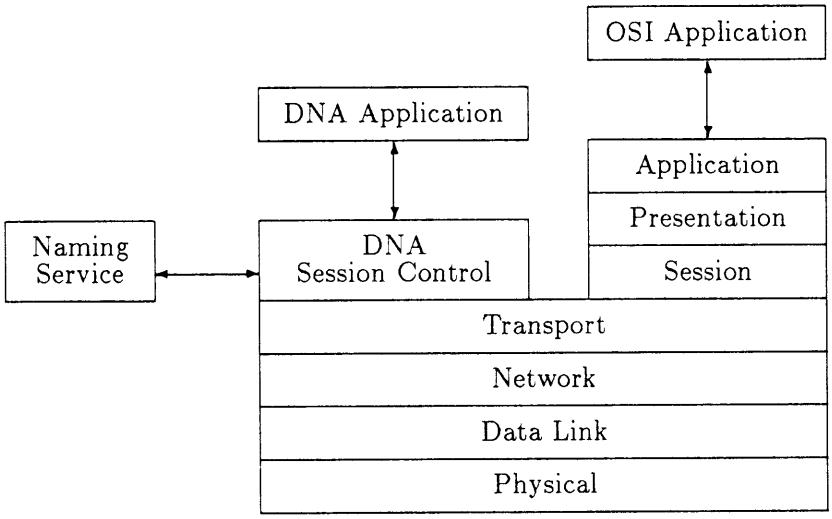


Figure 1.3: The Layers of DNA

Figure 1.3 shows the layers of DNA and their relationship. These are further described below.

Physical layer. The Physical layer is concerned with transmission and reception of data on the transmission medium. Its functions include:

- Conversion of data to and from electrical signals.
- Monitoring signals from the communication channel, such as modem signals.
- Bit level synchronization, either by direct generation and recovery of the clock signals or using an external clock (for example, from a modem).
- Management of Physical layer connections, such as dialup lines and X.21 circuits. The Physical layer provides the mechanisms for this but the policies (when to initiate and terminate a circuit) are administered by the Network layer.

The service provided by the Physical layer to its client is concerned with the transmission and reception of individual bits that make up higher-layer messages. In this layer, detailed operation is fully described by standards such as EIA RS-232-D, CCITT V.24 and X.21, and ISO 8802. DNA specifications are therefore concerned only with network management and with interfaces to other components of the architecture.

Data Link layer. The Data Link layer provides a dependable communication path between directly-connected systems in a network. It operates protocols which detect errors induced in the Physical layer (for example by electrical disturbances). This layer also corrects such errors where the anticipated error-rate is high, for example on synchronous lines.

DNA defines three protocols for this layer. *DDCMP*™ and *HDLC* operate over synchronous lines. DDCMP is a Digital-defined protocol used in previous Phases of DNA, which can also operate over asynchronous lines. HDLC uses selected features of the ISO standards. A subset of this protocol is compatible with *LAPB*, used in X.25. Over Local Area Networks, DNA uses ISO 8802-2 and -3, which are identical with the corresponding parts of IEEE 802, and can also use the Ethernet standards for compatibility with Phase IV.

Network layer. This layer routes user data between systems in the network. In DNA, this layer provides the *Connectionless-mode Network Service* (CLNS) of OSI. Each piece of user data (that is, each NSDU) travels through the network in a self-contained *packet,* containing its destination and source addresses and other required information. The protocol used is the ISO Internetwork Protocol (ISO 8473). This layer exists both in the communicating systems and also in the *routers* which join together the data links forming the network. The Network layer also maintains the databases needed to find routes through the network from a source to a destination. To make this possible, it operates a dynamic adaptive routing algorithm which automatically adapts the routes to the current topology of the network. This algorithm is defined in the *DNA Network Routing* specification. It allows for networks of several hundred thousand systems. The Network Routing specification also contains the mechanisms for using the various kinds of communications media, such as LANs, synchronous circuits and X.25 networks, and for interworking with other (non-DNA) networks that also use the OSI protocols.

DNA also provides the *Connection-mode Network Service* (CONS) in systems which are directly attached to an X.25 Wide Area Network, or which have access to such a network through the X.25 Gateway Access Protocol.

Transport layer. This layer provides a reliable end-to-end service between communicating systems, concealing from its users the detailed way in which this is achieved. This layer (and all higher layers) operates only in the communicating systems. Its functions include:

- Recovery from data loss, where the Network layer fails to deliver a packet (for example due to congestion), by retransmission.
- Flow control, to match the transmission rate to the reception rate.
- Congestion avoidance – using information provided by the Network layer, the Transport layer manages its use of the network to avoid saturation of the network while providing a fair service to all users.

- Segmentation and reassembly of user messages, so that there is no limit to the size of user messages (TSDUs) even though the size of network messages (NSDUs) is limited.

DNA contains two protocols for this layer. *NSP* (Network Services Protocol) is a Digital-defined protocol which was first defined for DNA Phase I. The *OSI Transport Protocol* (ISO 8073) is used for the first time in Phase V. Digital systems use Class 4 of this protocol, which can operate over both the Connectionless-mode and Connection-mode Network Services. DNA also describes the uses of Classes 0 and 2, for use only over the CONS.

Above the Transport layer, DNA permits two alternative modes of access. Proprietary and user-developed application protocols use the **DNA Session Control** layer. Standardized application protocols use the **OSI Session, Presentation and Application** protocols.

DNA Session Control. This provides the *logical links* (that is, connections) which are seen by the application programs. For data transfer, these provide the functions of the Transport service. Session Control's additional functions are concerned with the initiation of a logical link and its management within a system. The most important functions of this layer whose operation is specified by the architecture are:

Name to address translation. Remote objects are addressed by their *object name*. Session Control translates the name into the addresses used at each of the protocol layers, and in particular the Network address which allows the Network layer to send traffic to the correct remote system. For compatibility with Phase IV, objects may also be addressed using a *node name* which identifies the remote system, and a means to identify an *object* within the system.

Protocol selection. At several layers, DNA provides a choice of protocols. Partly, this is for compatibility between Phases, and partly it is to permit communication with non-DNA systems which have different choices in their selection of OSI options. When Session Control performs the object name translation, it also obtains information about the protocols which can be used. This is compared with the capabilities of the local system, and a mutually compatible protocol set is selected.

Access control. Session control provides features whereby remote users can be identified. In conjunction with system-specific access control mechanisms, this allows access to particular objects to be restricted to a limited set of remote users.

In addition, the Session Control layer is where system-specific logical link initialization functions are performed. These include process addressing and process activation.

OSI Upper Layer Protocols. The OSI Session, Presentation and Application layers are used to support standardized application protocols. DNA does not specify the operation of these layers, since they are fully described in the corresponding standards.

User access is in general provided at the application layer, although some implementations may provide direct access to the services of the Session and Presentation layers.

User access to DNA modules is not restricted to the DNA and OSI application layers. For example, there are several applications which make direct use of the Data Link layer, to meet specialized requirements in a LAN environment.

1.3.3 Message Flow in DNA

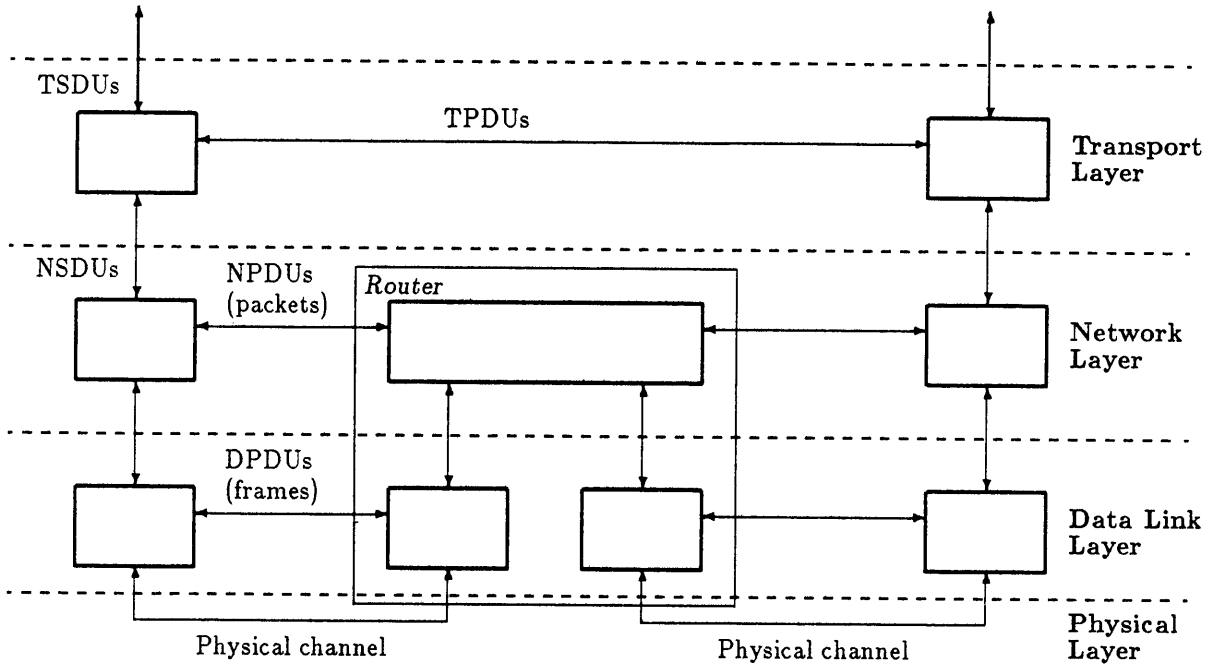


Figure 1.4: DNA Message Flow

Figure 1.4 shows how messages flow between DNA systems. At the Data Link layer, communication is strictly between directly connected systems. The messages passing between entities at this layer are Data Link PDUs (DPDUs), which are often called *frames*. At the Network layer, *routers* pass traffic from one data link to another on the path from one system to another. The messages at this layer are Network PDUs (NPDUs), often called *packets*. Routers examine the NPDUs to determine the next part of the route, and update some parts but largely pass them on unchanged. NPDUs passing between Network Layer Entities correspond to Network Service Data Units (NSDUs) which pass between the Network layer and the Transport layer.

At the Transport layer, only the end systems are involved. The messages passing between the Transport entities are Transport PDUs (TPDUs). They are not examined by the routers, but pass through transparently as user data. Data passing between Transport

and its users is in Transport Service Data Units (TSDUs). The user will normally be either DNA Session Control or the OSI Session layer.

1.3.4 The Naming Service

The name translation and protocol selection functions of the DNA Session Control layer require a large database containing information about all systems in the network. It would be an impractical task for each system to keep its own copy of this database, particularly in a large network. DNA therefore includes a network-wide *Naming Service*, which can store all names having network-wide significance. Any name can be accessed and its associated attributes (such as network addresses) can be retrieved, from any point in the network. Every name is unique. A hierarchical *directory structure* (similar to a filestore directory structure) simplifies the task of assigning unique names.

Names are physically stored in *directories*. In general, a directory is *replicated,* that is, stored on more than one physical system. This protects against hardware failure (for example, disk crashes) and also makes the information accessible even during temporary network problems. The Naming Service contains the mechanisms needed to ensure that the copies of directories become synchronized as changes occur.

A major use of the Naming Service is to maintain the database of object names, and associated addresses and protocol selection information, which Session Control uses. However, it is not constrained to this application, and can be used by any application which requires names to have unique network-wide significance.

The Naming Service is both a component of DNA and also a user of it. Communication between the *Name Servers* which provide the service uses the normal services of DNA through Session Control.

1.3.5 Network Management

DNA provides powerful distributed network management facilities which allow network components to be managed remotely. The network management facilities available include:

- Configuring the network components within a system, for example to determine how its physical links are to be used.
- Setting operational parameters such as timer values and resource limits.
- Examining configuration information and the operational parameters.
- Examining counters of diagnostic information such as messages sent and received and error conditions.
- Generation of events which can be collected and analyzed at a remote site, such as circuit failures and recovery.
- Diagnostic operations, in particular *loopback* of data to allow a failing component to be isolated.

Each system in the network, and each component (such as a protocol module) within a system, contains a network management *agent*. The agent accesses information held by the component, to report on it or to modify it. The information which is kept by each component, and how it may be modified and used, is specified by the corresponding part of DNA. Each system has a unique agent called the *node agent* which receives requests from other systems and dispatches them to the appropriate part of the system.

Network management is used through a *director*. DNA defines a network management command language called NCL which corresponds closely to the architected structure of a system. A program which provides an NCL interface to a user is an example of a director. A director may either be local (that is, running on the system it is managing) or remote. A local director can be used to perform management operations when the system is not yet part of a network, for example during system initialization. A remote director communicates with the system it is managing using the following protocols.

Common Management Information Protocol (CMIP). This is used for communication between a director and the target node agent. It provides an encoding of the network management operations which can be performed, and their parameters. For example, an operation might be "read counter" and the parameters would be the name of the counter and the name of the system component in which it resides. CMIP is a client of the Session Control layer, and operates in the same way as any other DNA application protocol.

Maintenance Operations Protocol (MOP). This is used for low-level communication when a system is not fully operational. It is a client of the Data Link layer. Its most important functions are the triggering and execution of down-line load and up-line dump, and link-level loopback functions.

Event Logger Protocol. This is a particular form of CMIP, used by the node agent to report events when this has been requested by a director. Like CMIP it operates as a DNA application protocol.

Network Information and Control Exchange (NICE). This is the protocol used by Phase IV for management communication. It is supported by NCL for management of Phase IV systems during the migration period.

1.4 Relationship of DNA to OSI

As Figure 1.3 shows, the layers of DNA are in direct correspondence with the seven layers of the OSI Reference Model (ISO 7498). This reflects both the origin of OSI, which derived from contemporary network architectures including DNA, and the continuing development of DNA to align it more closely with international standards. In Phase V, the protocols used up to the Transport layer are the OSI protocols, earlier proprietary protocols being retained as an alternative for compatibility purposes. Specifically, the following major OSI standards are used:

Physical layer. The DNA Physical layer has always been based on available standards, which are implemented in hardware devices. These include, EIA RS-232-D (and the corresponding ISO standards and CCITT Recommendations) and the Ethernet standards as now reflected in ISO 8802-3 and IEEE 802.3.

Data Link layer. The version of HDLC which is used in DNA uses ISO 4335 and ISO 7809. For Local Area Networks, DNA uses the Logical Link Control protocol of ISO 8802-2 and IEEE 802.2.

Network layer. The ISO *Protocol for Providing the Connectionless-mode Network Service,* ISO 8473, is used for data transfer. Exchange of routing information between end-systems and routers uses the ISO *End System to Intermediate System Routing Protocol,* ISO 9542. The Connection-mode Network Service, where used, is provided using the X.25 Packet Layer Protocol, as defined in ISO 8208 and the mapping defined in ISO 8878. The addresses used by the DNA Network layer follow the corresponding OSI standard, ISO 8348 Addendum 2.

Transport layer. DNA includes Class 4 of the OSI Transport Protocol, ISO 8073. Classes 0 and 2 are also available, for use over the CONS. The service provided to a DNA Application corresponds to the OSI Transport Service, ISO 8072. This service is also used by applications which use the OSI upper layer protocols.

Above the Transport layer, OSI protocols are available as an alternative but are not used for all applications. There are several reasons for this:

Functions available. DNA supports many applications and their protocols. Only a few of these are under study for OSI standardization.

State of the standards. At the time of writing, the only OSI upper layer standards which have been approved as an International Standard are those for the Session layer. The others are at various stages of drafting, and are therefore subject to further change before final approval.

Compatibility. Communication with previous implementations is essential and can only be done using the proprietary protocols.

In Phase V of DNA, specific implementations of upper layer standards will be included as they become available. Development of an integrated DNA architecture for the OSI upper layers will take place during the lifetime of Phase V, reflecting the progress of the standards themselves.

The DNA network management protocols are based closely on early drafts of ISO's work on the subject. The CMIP protocol used in DNA is based on drafts of the ISO CMIP protocol (ISO DP 9596), and uses the OSI Remote Operations Service (ISO DP 9072) and the OSI Presentation protocol (ISO 8823 and ISO 8825).

The functions of the DNA Session Control layer are not yet specified for OSI. When they are, it is expected that they will reside in the OSI Application layer. Integration of these functions in the DNA use of the OSI Application layer will form part of future extensions to DNA.

1.5 Relationship to Phase IV

In keeping with the goals of DNA, a system which implements Phase V is fully able to communicate with a Phase IV system. A Phase IV network can be migrated gradually to Phase V without requiring closely synchronized changes to multiple systems. All of the protocols used in Phase IV are included in Phase V systems, although in many cases they will be used only for communication with Phase IV systems. To the user of a DECnet network, the change from Phase IV to Phase V will not be apparent, except that the new functions of Phase V will gradually become available.

A summary of the major changes between Phase IV and Phase V is given below.

Terminology. The names of some layers have changed from Phase IV:

- The *Network* layer was called the Routing layer.
- The *Transport* layer was called the End Communications layer.
- The *Physical* layer was called the Physical Link layer.

These changes have been made for alignment with the names used in OSI.

Local Area Networks. Phase V uses the frame format defined by ISO 8802-2 and -3 and IEEE 802.2. This differs slightly from the Ethernet frame format used in Phase IV. DNA systems automatically use the correct format for communication with other systems on the same LAN. A Phase V system always transmits and receives in ISO format, and also listens for frames in Ethernet format. If there are Phase IV systems on the LAN (that is, if Ethernet format frames are received) then it also transmits using Phase IV format when necessary.

Data Link protocols. Phase V supports HDLC as well as DDCMP. A circuit must be configured to use the same protocol as the remote system, otherwise it will fail to initialize.

Network layer. There are many changes to the Network layer in Phase V, to accommodate the OSI standards and to permit larger networks. These are described in Chapter 5.

Transport layer. Phase V supports the OSI Transport Protocol as well as NSP. A common interface is presented to the user of the Transport layer. Selection of the correct protocol, for communication with another DNA system, is performed by the Session Control layer. For communication with non-DNA systems, the OSI Transport Protocol is always used.

Session Control layer. The major change to this layer is the use of the Naming Service for translation of object names and obtaining protocol selection information. Application programs which use this layer (which is the layer normally exposed in DECnet products) need not change in Phase V, but some new facilities available through the Naming Service will only be available if corresponding new facilities of the interface are used.

Network Management. Both the protocol and the command language (NCL) are new for Phase V. The Phase IV command language (NCP) and protocol (NICE) are available for communication with Phase IV systems.

Chapter 2

Physical Layer

The Physical layer provides mechanical, electrical, functional and procedural means to activate, maintain and deactivate *physical connections* between directly connected systems, and for Physical layer clients to transfer data. Functions of the layer include encoding and decoding signals on the physical interface, bit synchronization of received data, bit transmission, and interfacing the communications channel to the processor and memory used to implement higher layer protocol functions. Implementations of this layer encompass hardware interface devices and device drivers in operating systems, as well as communications hardware such as modems, transceivers, and the physical communications links themselves.

Protocols for the Physical layer are rudimentary. No special Physical layer standards have been developed for DNA. Instead, it relies on industry standards for the Physical layer, thereby ensuring that DECnet products can operate over available technologies and physical networks. Physical layer standards supported by DNA for wide area networks include the EIA RS-232-D and RS-423 specifications, and the CCITT V.24 and X.21*bis* specifications. Physical layer standards supported by DNA for Local Area Networks include two baseband implementations of the ISO 8802-3 LAN, the original and the ThinWire™ specifications, and a broadband implementation.

Phase V of DNA defines the functions and operation of the Physical layer more completely than earlier versions of the architecture. The way in which industry standards for the layer are integrated into DNA is fully described. Network management of the Physical layer, which in Phase IV was bound to the management of Data Link layer protocols, has been rationalized and enhanced.

DNA support for CCITT Recommendation X.21 has been added to Phase V, allowing implementations of DECnet access to and use of the facilities of X.21 circuit switched public data networks. The architecture for X.21 has been closely aligned with that for modem connect type lines, such that a common set of services is presented to the higher layers of DNA.

The DNA Physical layer is structured in a modular way, allowing support for other Physical layer standards, such as those for ISDN, to be added in later versions.

2.1 Physical Layer Functions

The following functions are performed by Physical layer modules.

Connection establishment and release. These functions are provided where a communications link is attached to a circuit switched wide area network such as an X.21 or public telephone network. They allow physical connections to be dynamically established and released.

Bit synchronization. This function establishes synchronization with an incoming bit stream, and thereafter clocks data in from the communications channel at the correct rate.

Data transfer. The Physical layer provides a bit stream service interface where data is passed across the interface in 1-bit units. On the transmit side, data bits passed by the client are clocked onto the communications channel. Conversely on the receive side, bits are clocked in from the channel and presented to the client. The Physical layer maintains sequence; data bits are delivered in the same order in which they were submitted. A communications link may allow full-duplex or half-duplex transmission.

Fault notification. This function reports fault conditions to Physical layer clients.

Management. These functions allow network management to control and monitor the operation of the Physical layer, for example to set operating characteristics of communication links, activate and deactivate physical connections, monitor the status of active connections and perform loopback. Physical layer modules maintain management counters and report significant events.

2.2 Physical Layer Functional Modules

The DNA Physical layer contains a number of functional modules, one for each type of communications interface supported by DNA. These modules interface with the communications channel and make Physical layer services available to the higher layers of DNA. Many of the functions performed at the Physical layer are common across all the modules although details vary in each case. The modules comprise functions to get and set DTE-DCE interchange circuit signals, where applicable, and functions to transmit and receive data. At this level, internal system requests are mapped onto electrical signals on the communications channel. These functions are media specific. For example, the transmit function for a CSMA/CD LAN differs from that for an X.21 interface.

Currently, the DNA Physical layer defines functional modules for the following communications interfaces:

- **Modem Connect**. This is described in Section 2.3.
- **X.21**. This is described in Section 2.4.
- **CSMA/CD LAN**. This is described in Chapter 4.

2.3 Modem Connect Functional Description

DNA Modem Connect is the name given to the Physical layer module which defines how DNA operates over communication links conforming to industry standards for modem connection. There are a number of these standards, each with different characteristics. However, the standards encompassed by DNA Modem Connect are sufficiently similar to be covered by a single architectural module. Differences between the standards are treated as variants within this general module for modem connection.

DNA Modem Connect defines how the Physical layer services provided by switched and leased lines are integrated into DNA. It defines the mechanisms which allow higher layers of DNA to use these services, and for network management control and monitoring of lines. The module supports a number of the industry standards for the layer, including EIA RS-232-D and CCITT V.24, and CCITT V.25 and V.25*bis* for auto call and auto answer. The module also defines *null* modem operation, where only the data interchange circuits are used.

2.3.1 DNA Modem Connect Features

This section describes some of the major features of DNA Modem Connect.

2.3.1.1 Call Control Services

DNA Modem Connect defines services which permit the control and monitoring of switched line connection establishment and release. This allows clients to make outgoing calls, handle incoming calls and clear calls. The progress of an outgoing call attempt may be monitored, and reason information is returned if the call fails to connect. Call control services do not apply to leased line connections.

2.3.1.2 Data Transfer Services

DNA Modem Connect defines services which permit the higher layers of DNA to transmit and receive data on leased and switched lines, and to control line turnaround operations on half-duplex communications links. These services are used by the Data Link layer.

2.3.1.3 Call References

DNA Modem Connect uses *call references* to identify calls locally. Call references are values assigned by DNA Modem Connect to outgoing call attempts placed to the network and incoming calls which successfully connect. Each value is unique, allowing every call (attempt) to be unambiguously identified. By identifying calls in this way, a number of additional features are possible. In particular, call references provide:

- A means to tie together the call signaling (call establishment and release) and data transfer phases of a call, avoiding potential problems of mis-synchronization between the phases.

- A means by which management information for the call signaling and data transfer phases can be correlated by network management.

2.3.1.4 Call Sharing

DNA Modem Connect provides a feature which permits calls on a switched line to be accessed by more than one client. This is known as *call sharing*. Call sharing allows, for example, DNA MOP and the Network layer to utilize the same line. When call sharing is used, clearing is coordinated by DNA Modem Connect such that a request by one client to clear a call does not have immediate effect if it is still in use by others.

2.3.1.5 Network Management

In Phase V, a **Line** management *entity* is defined which contains Physical layer management attributes. These include speed attributes to control primary and fall-back transmission rates, type attributes which define line and modem capabilities (null modem, switched or leased line), a communications mode attribute denoting asynchronous or synchronous operation and an attribute to control the transmission encoding technique used, for example, NRZI encoding.

2.4 X.21 Functional Description

X.21 is a Recommendation of the International Telegraph and Telephone Consultative Committee (CCITT). It describes the electrical, mechanical and procedural interfaces between *Data Terminal Equipment (DTE)* and *Data Circuit-terminating Equipment (DCE)* for operation over synchronous public data networks.

X.21 defines two types of service: leased circuit and circuit switched. The leased circuit service provided to higher layers is equivalent to that provided by more traditional leased line services. In its simplest form, two DTEs attached to the network are connected point-to-point. The circuit switched service is akin to that provided by the telephone network. A DTE attached to the network controls the establishment and release of physical connections to other DTEs. The establishment and release procedures are defined by X.21, with the call set-up phase involving the exchange of address and facility data with the network. Higher layer data is carried transparently over leased and established circuit switched physical connections.

DNA X.21 defines how the services provided by X.21 networks are integrated into DNA. It defines the mechanisms which allow higher layers of DNA to use these services, and for network management control and monitoring of subscriber lines. DNA X.21 supports leased circuit and circuit switched connections.

2.4.1 DNA X.21 Features

This section describes some of the major features of DNA X.21.

2.4.1.1 Call Control Services

DNA X.21 defines services which permit the control and monitoring of X.21 circuit switched connection establishment and release. This allows clients to make outgoing calls, handle incoming calls, access call information and clear calls. Call control services do not apply to leased circuit connections.

Clients can request the use of outgoing call facilities such as reverse charging or closed user groups on a per-call basis. CUG and other facilities, if available from the network, may be used to set up incoming call *selection criteria* which limit the incoming calls reported to a client to those matching the criteria, for example to calls specifying a particular CUG.

Call control services allow a client to monitor the progress of an outgoing call attempt and obtain reason information from the network if the call fails to connect. When a call connects, the client can request full details of the call as it was actually established, for example if it was redirected and if so, where to.

2.4.1.2 Data Transfer Services

DNA X.21 defines services which permit the higher layers of DNA to transmit and receive data on leased circuit and circuit switched connections, and to control half-duplex operations where these are supported by the network. These services are used by the Data Link layer.

2.4.1.3 Call References

DNA X.21 uses the call reference mechanism as described in Section 2.3.1.3.

2.4.1.4 DTE Sharing

DNA X.21 provides a feature which permits a subscriber line to be shared by more than one client. There are two variants of this: *call sharing,* in which many clients are permitted access to the same call, and *line sharing,* in which only one client has access to any single call. DTE sharing allows, for example, DNA MOP and the Network layer to utilize the same subscriber line.

Line sharing is only possible if the network provides Closed User Group and/or *sub-addressing* facilities enabling different selection criteria to be set for each client. When call sharing is used, clearing is coordinated by DNA X.21 such that a request by one client to clear a call does not have immediate effect if it is still in use by others.

2.4.1.5 Network Management

DNA X.21 provides a comprehensive set of network management features allowing control of subscriber lines, use of optional facilities, selection of operating procedures and the definition of outgoing call parameter and call selection parameter profiles or *templates.*

DNA X.21 allows local and remote loopback testing to be performed, and on-line registration and cancellation of network facilities where this is supported by the network. This

allows, for example, the direct call number to be set or incoming calls to be redirected to another DTE.

Network management counters are maintained for each subscriber line, and significant events are logged such as DTE time-out, indicating incorrect network operation, and calls rejected.

Chapter 3

Data Link Layer

The Data Link layer provides means to establish, maintain and release *data link connections* between two directly connected systems, and for Data Link layer clients to transfer data. Modules in the Data Link layer co-operate to provide this logical communications path by use of a *data link protocol.* A data link protocol defines formatting rules and procedures for data transmission and reception over a physical communications link. In the absence of errors from the communications link, the task of the data link protocol is relatively simple. Once errors occur, however, data corruption and loss are possible, introducing synchronization problems between the transmitter and receiver. A data link protocol designed to operate under these conditions has error detection and recovery procedures. Data Link layer clients are notified of persistent or unrecoverable errors.

Data Link layer protocol modules use the services of the Physical layer described in Chapter 2 to transmit and receive data on communications links. Physical layer services to control operation of the link are also used. For example, when operating over a half-duplex modem connect line, a Data Link layer module uses Physical layer services to control line turnaround between transmit and receive modes.

3.1 Data Link Layer Functions

The following functions are performed by the Data Link layer.

Connection establishment and release. These functions are provided where the Data Link layer module supports a *connection mode* type of operation. They allow synchronization of data link operation with a peer module prior to data transfer, a process called data link connection, and termination of the connection when it is no longer required.

An alternative mode of operation called *connectionless* is supported by some Data Link layer modules. In this case, there are no establishment and release phases; the recipient of data is determined from addressing information supplied by the client with each service data unit.

Data transfer. User data is passed between the Data Link layer and clients in service data units. On the transmit side, this user data is *enveloped* with protocol control information before being sent. On the receive side, the control information is extracted before presentation of the data to the client. There is a one-to-one mapping of service data unit to protocol data unit.

Framing and synchronization. The synchronization functions assemble received Physical layer service data units (bits) into larger units (bytes and messages). The framing function detects the beginning and end of protocol messages, once synchronization has occurred.

Sequence control. This function uses sequence numbers in the protocol to ensure that the sequential ordering of data transmitted on the data link connection is maintained. It does not apply to connectionless operation.

Error detection. This function detects transmission, format and procedural errors on the data link connection.

Error recovery. This function attempts to recover from transmission, format or procedural errors. Persistent or unrecoverable errors are reported to the Data Link layer client. It does not apply to connectionless operation.

Flow control. This function limits the rate at which data is sent or received on the data link connection. It does not apply to connectionless operation.

Identification and parameter exchange. This function enables the exchange of identification and operational parameters over the data link connection, typically prior to user data transfer. This function may include parameter negotiation.

Use of Physical layer services. The Data Link layer uses the services of the Physical layer to transmit and receive data, and where applicable, to control the operation of the communications link.

Management. These functions allow network management to control and monitor the operation of the Data Link layer, for example to set data link protocol operating characteristics, enable and disable data link connections and monitor the status of enabled connections. Data Link layer modules maintain management counters and report significant events.

3.2 Data Link Layer Protocol Modules

Currently, the following protocol modules are defined in the DNA Data Link layer:

- **Digital Data Communications Message Protocol (DDCMP).** This is described in Section 3.3.
- **High-Level Data Link Control (HDLC).** This is described in Section 3.4.
- **CSMA/CD LAN.** This is described in Chapter 4.

3.3 DDCMP Functional Description

DDCMP was designed in 1974 by Digital specifically for DNA. It is a general-purpose, byte-oriented data link protocol which is designed to serve the needs of inter-computer data communications in a wide variety of applications and environments.

DDCMP operates over synchronous or asynchronous, switched or non-switched communications links. It can operate in *point-to-point* configurations or in *multipoint* configurations in which communication takes place between a *control* station and each of several *tributary* stations. DDCMP messages are framed as sequences of bytes, beginning with a single control byte indicating the message's starting point and type (data or control). While DDCMP control messages have a fixed length, data messages have variable lengths, indicated by a length field. On reception, this encoding allows the receiver to determine the beginning and end of messages. The encoding also allows user data transparency; data containing bit patterns that are those of control characters is not misinterpreted by DDCMP. Incoming bits are assembled into bytes using start/stop bits for asynchronous links and synchronization characters for synchronous links.

DDCMP uses two 16-bit *cyclic redundancy checks* (CRC-16); one to detect errors in the protocol header, and a second to detect errors in user data. On half-duplex or multipoint links, DDCMP executes link allocation procedures to ensure that two or more stations do not conflict in their use of the link. These techniques are based on *polling* in which one station gives permission to the other to transmit. DDCMP uses timers and sequence numbers to detect and recover from lost messages; it also prevents the process of error recovery from creating duplicate messages. The Network layer uses the error detection and retry capability of DDCMP to verify that links between directly connected systems are operational and to synchronize the operation of the routing protocols.

DDCMP makes efficient use of high speed communication links, such as satellite links, to provide high data throughput.

3.3.1 DDCMP Features

This section describes in more detail some of the major features of DDCMP and how they affect the operation of the protocol.

3.3.1.1 Link Management

Link Management resolves transmit contention on half-duplex point-to-point and multipoint links. In the half-duplex point-to-point case, one station transmits while the other receives. The switching between transmit and receive modes is accomplished by a *selection flag* in DDCMP messages. The transmitting station sets the flag in the last message sent, indicating that the receiver should enter transmit mode following reception of the message.

In the multipoint case, the link appears as a party line with one station designated the control station and all others tributaries. Each tributary is assigned an address using network management. Messages sent by the control station contain the address of a tributary. Tributaries ignore received messages unless they are the addressee.

Tributary stations transmit only when selected to do so by the control station. Like the half-duplex point-to-point case, this is done using the selection flag. Messages sent by a tributary to the control station are seen and ignored by all other tributaries. The control station receives the message and verifies that it has come from the selected station. Tributary stations may not exchange data directly; all traffic is between the control station and the tributaries.

3.3.1.2 Message Exchange

The message exchange feature of DDCMP is that part of the protocol which provides sequential, error-free data transfer. DDCMP is a *positive acknowledgment retransmission protocol.* For each data message correctly received and passed to the client, a positive acknowledgment is returned on the link to the transmitter. If incorrectly received, the message is not passed to the client. Instead, an error recovery mechanism is invoked causing message retransmission. DDCMP uses the CRC-16 cyclic redundancy check for error detection on protocol headers and user data.

The basic message exchange mechanism uses data message sequence numbers, a positive acknowledgment control message (ACK) and a timer. However, DDCMP has additional control messages and operational techniques to achieve higher performance and faster error recovery.

Negative acknowledgment (NAK). The duration of a timer used to detect an error when an ACK is not returned, must be long enough to account for propagation delay, line turnaround and processing of the data message and ACK generation by the receiver. Typically, values of a few seconds will be used. However, DDCMP uses a negative acknowledgment mechanism which, for most error conditions, enables retransmission to take place before the timer expires, making more efficient use of link resources. This mechanism is used by the receiver when it does not correctly receive a message but does recognize that one was sent to it. Under these conditions, a NAK control message is returned, initiating retransmission.

Reply to message number (REP). If the timer expires, this probably means that the returned ACK was lost or corrupted. In this case, rather than retransmit perhaps a large amount of data, a REP control message is sent containing the number of the previous message. This forces the transmitter and receiver to synchronize their numbering and to cause retransmission if required. If the message with that number was received correctly, the response to REP is an ACK; otherwise it is a NAK. NAK causes retransmission to occur.

Pipelining. *Pipelining* is a technique which allows further messages to be sent while awaiting acknowledgment of ones previously transmitted. To achieve this, DDCMP messages are numbered from 0 to 255. The numbering is *cyclic (modulo 256),* meaning that after message number 255 the next number used is 0. ACKs not only confirm that the specified message has been received correctly, but also all other unacknowledged messages since the previous ACK. If an ACK is lost or corrupted, the acknowledgment information is automatically included in subsequent ACKs.

Piggybacking. The purpose of an ACK is to convey the message number of the last successfully received data message. If data traffic is being transmitted in both directions, this message number can be carried in a message sent in the reverse direction. This technique is called *piggybacking,* and saves the framing overhead for a separate ACK message.

ACK implied in NAK. Like an ACK, a NAK contains the message number of the last message successfully received. Therefore, as well as indicating an error, receipt of a NAK implies acknowledgment of previous messages.

Initialization. The process of setting message numbers to initial values is called *initialization.* It is accomplished by use of STRT (Start) and STACK (Start Acknowledge) control messages. Initialization is carried out at link start-up time or after a failure, to reset the message numbers at both ends of a data link connection to zero. The protocol is designed so that one end of a connection cannot be initialized without the other.

3.3.1.3 Maintenance Mode

In maintenance mode, DDCMP provides a connectionless service to the DNA Maintenance Operations Protocol (MOP). In this mode, DDCMP framing and link management features are used but not the message exchange features described in Section 3.3.1.2. Sequence number management and message acknowledgment, if required, must be handled by MOP itself. A data link connection cannot be used to carry normal DNA traffic while in maintenance mode. MOP is described in Section 10.2.3.

3.3.1.4 Network Management

The operation of DDCMP is unchanged from Phase IV. Management of the DDCMP module, although largely unchanged in function, has been upgraded in accordance with the Phase V network management model. In Phase IV, the Line database contained both Data Link layer protocol and Physical layer management parameters. In Phase V, management of Data Link layer protocols is separated from Physical layer management. For each communications link on which DDCMP is to operate, a **Link** management entity is created. This contains the data link management attributes *common* across all DDCMP stations on the line, such as duplex mode and protocol type. Additionally, each Link entity has one or more **Station** management sub-entities, where each Station contains data link management attributes *specific* to operation with a single DDCMP station on the line, such as station address and protocol state. In point-to-point and multipoint tributary configurations, each Link entity has a single Station sub-entity. In multipoint control configurations each Link entity has up to 255 Station sub-entities; one for each tributary controlled by the protocol.

The Link and Station management entities contain the data link protocol related attributes which in Phase IV were held in the Line and Circuit databases respectively.

The 1-bit DDCMP counters of Phase IV have been replaced in Phase V by larger counters. A more comprehensive set of management events has been defined.

3.4 HDLC Functional Description

HDLC is a bit-oriented data link protocol which operates over synchronous, switched or non-switched communication links. Its various aspects are described in a series of standards including:

- ISO 3309 — HDLC frame structure
- ISO 4335 — HDLC elements of procedure
- ISO 7809 — HDLC classes of procedure
- ISO 8471 — HDLC data link address resolution
- ISO 8885 — HDLC general purpose XID information field content and format.

In addition, a subset of HDLC known as *LAPB*, which is used in X.25 level 2, is defined in:

- ISO 7776 — X.25 DTE data link layer
- CCITT Recommendation X.25 (1984)

Taken together, these standards define a whole range of protocols, procedures and features, many of which are not capable of inter-operation. The DNA specification of HDLC uses a subset of these, selected to provide the characteristics and features required by DNA and to permit inter-operation with other common subsets.

3.4.1 HDLC Features

The following sections describe in more detail, some of the major optional features of HDLC and the way they are used in DNA HDLC.

Operational modes. The HDLC standards specify various operational modes. In DNA HDLC two modes are supported. *Normal Mode* is used only over a half-duplex physical line, whereas the more efficient *Balanced Mode* is used over a full-duplex line. A DNA HDLC station in Normal Mode can operate as a primary on a point to point physical link, or as a secondary on either a point to point or multipoint physical link.

Extended sequence numbering. In the base HDLC standard, sequence numbers are modulo seven, but when the optional *extended sequence numbering* is in use, this is increased to 127. This is required for efficient operation over physical links which have a large propagation delay in relation to their bandwidth (notably satellite links). DNA HDLC will normally be configured to use extended sequence numbering.

Error detection. HDLC uses a single cyclic redundancy check to detect errors in both the header and data fields. This may be either a 16-bit or 32-bit CRC. DNA HDLC uses a 32-bit CRC by default in order to provide maximum protection against undetected

errors, but it will use a 16-bit CRC when communicating with a station which does not support the use of a 32-bit CRC. Additionally, when HDLC is being used between two DNA stations, a user data length field is included in the user data.

Negotiation of options. Many of the operational parameters of HDLC, such as station address, window size or the use of various options, must be set correctly before two stations can communicate. A special frame, the *XID* frame, may be exchanged during link initialization to allow these to be negotiated to mutually acceptable settings. Support for the XID negotiation procedures is included in DNA, but it is defined to be optional by the ISO standards. When communicating with a remote non-DNA station which does not support XID negotiation, the stations must be manually configured via network management.

Maintenance operations. Unlike DDCMP, there is no special Maintenance Mode in HDLC. Such messages are instead carried in *UI* (Unnumbered Information) frames. These may be exchanged even when the sequenced data mode has not yet been initialized. Since UI frames are not subject to error correction, sequencing or flow control, such functions are handled in DNA by the higher layer management protocol (MOP).

Chapter 4

Local Area Networks

Local Area Networks (*LANs*) are a type of communication facility that provides high speed communication in a moderate size geographical area. Phase V of DNA supports the CSMA/CD (Carrier Sense Multiple Access with Collision Detect) class of LAN, as defined by the ISO 8802-3 and Ethernet standards. In the ISO Reference Model, Local Area Networks reside in the Physical and Data Link layers. For example, the ISO 8802-3 standard defines both Physical layer aspects such as the type of cables to be used, and Data Link layer aspects such as frame formats and services provided to higher layers.

4.1 Services Provided

The DNA CSMA/CD Data Link module provides the following services:

Transmission and reception of datagrams. The LAN Data Link module detects data corruption, but does not guarantee delivery of all frames. This type of Data Link service is a *datagram* service.

Support for multiple stations. A LAN may have thousands of stations; each has a unique address. Datagrams specify both source and destination addresses.

Multicast. Datagrams may be sent to a *multicast* (*group*) address to send to a group of related stations with a single frame.

Multiple protocols. The LAN Data Link module allows for multiple protocols to be used on the same LAN, and on the same station.

Shared access. Because the LAN is shared among all the attached stations, each station controls its access to the LAN to ensure fair and efficient use of the shared resource.

Management. The LAN Data Link module provides a management interface for monitoring of usage, diagnosis and control.

4.2 Characteristics of the CSMA/CD LAN

Some of the characteristics of the CSMA/CD LAN are:

- Physical layer using coaxial cable segments of up to 500 meters length, or ThinWire™ cable segments of up to 185 meters length.
- Segments may be interconnected by repeaters into a LAN. The resulting LAN must have a tree topology (without loops), and there may be at most two repeaters between any two stations. There may be up to 1024 stations on a LAN, with a maximum distance between any two stations of 2.8 kilometers.
- Channel bit rate of 10 million bits per second.
- Frame size in the range 64 to 1518 bytes.
- Sharing of access by *carrier sense* and *collision detection*. Carrier sense means each station waits for the channel to become quiet before transmitting. If two stations start transmitting simultaneously, this produces a *collision*, which is detected and causes the transmission to be retried as needed. The result is a Data Link which offers very low delay at low to moderate loads, and degrades gracefully under high loads or overloads, without instability. With this simple mechanism, the CSMA/CD LAN provides fair service to all stations regardless of load.
- LANs may be interconnected by *bridges*, thereby creating an Extended LAN. Refer to Section 4.5 for more detail.

4.3 Relationship to International Standards

The ISO 8802 series of standards define a number of Local Area Networks. (These standards were originally developed by IEEE; they have since become ISO standards.) Phase V DNA supports the CSMA/CD class of LAN, as specified by ISO 8802-3 (IEEE 802.3). It also supports the Ethernet standard.

LAN Bridges are the subject of IEEE 802.1, part D (*MAC Bridges*) and ISO 8802-1. Significant portions of these standards are based on the architecture of DNA Bridges. Since these standards are still in a draft stage, conformance to the eventual standards is not yet assured but migration will be greatly eased.

4.3.1 Addressing

Local Area Networks support connection of multiple stations, so each frame has to specify which station(s) should receive it. The ISO 8802 standards specify 48-bit *addresses* for this purpose. Each frame contains both a destination address and a source address.

The first two bits in the address (Figure 4.1) have a special function. The first bit is the Individual/Group address bit. If this bit is set, the address is a *group* (*multicast*) address.

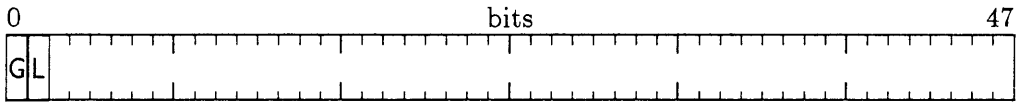


Figure 4.1: LAN address format

If this bit is clear, the address is an *individual* station address. The source address must be an individual address.

The second bit is the Globally/Locally administered address bit. If this bit is clear, the address is from an address block assigned by the IEEE. If this bit is set, the address is from a locally managed address block. For example, addresses beginning with the 24-bit block identifier 08-00-2B are assigned to Digital, and Digital assigns individual unique addresses from that block to its products. When locally managed addresses are used, it is the responsibility of the local administration to ensure that addresses are unique.

Multicast addresses are used to send a single copy of a frame to a number of stations. The multicast address designates a particular class of station (and LAN Data Link user) rather than a specific station. For example, the Network layer on DNA Level 2 routers receives frames addressed to the multicast address 09-00-2B-02-00-00 (see Section 5.2.2).

4.3.2 Multiplexing on ISO 8802-3 LANs

The LAN Data Link module allows multiple Data Link users on the same system (multiple higher level protocols) to access the LAN concurrently. This is done using the procedures and frame formats defined in ISO 8802-2 (IEEE 802.2), *Logical Link Control* (*LLC*). DNA uses LLC Class 1, which provides a datagram service. The standard also defines Class 2, which is not used by DNA protocols. The general format of an LLC frame on the ISO 8802-3 LAN is shown in Figure 4.2.

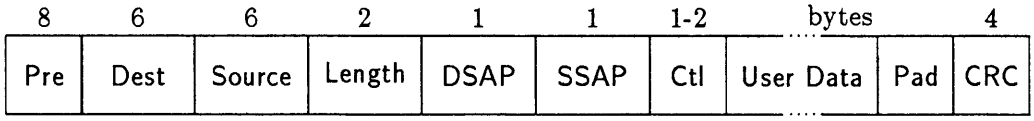


Figure 4.2: ISO 8802-3 frame format

Description of the fields:

Pre. Preamble and starting delimiter. This is the bit pattern 1010 ··· 1011, which the CSMA/CD Physical layer uses to locate the start of the frame.

Dest. 48-bit destination address.

Source. 48-bit source address.

Length. The number of bytes in the DSAP, SSAP, Ctl and User Data fields.

DSAP. Destination Service Access Point address.

SSAP. Source Service Access Point address.

Ctl. LLC Control field. For LLC Class 1 this is one of:

- **UI.** Unnumbered Information (datagram) LLC frame.
- **XID.** Exchange Identification LLC frame.
- **Test.** Test (loopback) LLC frame.

User Data. Data supplied by the LAN Data Link user.

Pad. Pad bytes used, if necessary, to fill out the frame to a minimum total size of 64 bytes.

CRC. The ISO 8802-3 32-bit Cyclical Redundancy Check error detecting code.

In the same way that destination and source address identify the receiving and sending station, the destination and source SAP (DSAP and SSAP) addresses identify the receiving and sending LAN Data Link user. The values for the SAP address fields are assigned by the IEEE; each value identifies a particular protocol defined by a national or international standard. For example, the SAP address assigned to the ISO Connectionless-mode Network protocol (ISO 8473) is FE (01111111).

Since SAP addresses are assigned only to standardized protocols, another frame format is used for protocols that are not national or international standards. This frame format is known as *Subnetwork Access Protocol* (*SNAP*) and is illustrated in Figure 4.3. It is a variant of the UI frame of LLC Class 1, with SSAP and DSAP both set to the SAP address for SNAP, which is AA.

The five bytes following the Ctl field are the *Protocol ID*, which is used to designate a particular LAN Data Link user. The first three bytes are taken from the 24-bit block identifier assigned by IEEE (see Section 4.3.1); the last two bytes are assigned by the owner of the block. For example, 08-00-2B-80-3C is the Protocol ID for the DNA Naming Service, advertising availability of Name Servers on a LAN. As with station addresses, the first two bits of the Protocol ID have special significance. The first bit is reserved and must be zero. The second bit is zero for globally assigned Protocol IDs, and one for locally assigned Protocol IDs.

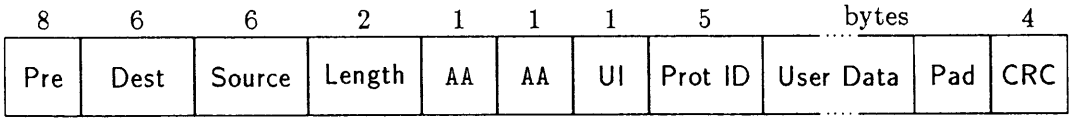


Figure 4.3: ISO 8802-2 SNAP frame format

4.4 Ethernet

Ethernet is an earlier standard for a CSMA/CD LAN very similar to ISO 8802-3. It uses the same Physical layer as ISO 8802-3, and the same format 48-bit addresses and CRC. Multiple protocols are supported differently than in LLC. Figure 4.4 illustrates the Ethernet frame format.

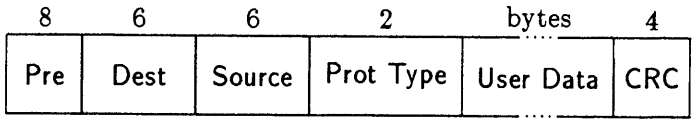


Figure 4.4: Ethernet frame format

The Ethernet frame format uses the two-byte Protocol Type field to identify the LAN Data Link user. The Protocol Type values are assigned by the Ethernet address administration. For example, 90-00 is the protocol type for the Ethernet loopback protocol.

Unlike the ISO 8802-3 frame format, the basic Ethernet frame format does not include a length field. The CSMA/CD Physical layer requires that all frames must be at least 64 bytes long (not including the Pre field). In the basic Ethernet frame format, the LAN Data Link user is required to ensure this by always supplying at least 46 bytes of user data for each frame (attempts to supply less data are rejected by the LAN Data Link module).

To eliminate the need for each Data Link user to deal with the minimum CSMA/CD frame size, the DNA CSMA/CD Data Link module also supports a variant of the basic Ethernet frame format, which includes a length field and padding. This format is shown in Figure 4.5. There is no indication in the frame header to distinguish the padded format from the basic format; it is up to the communicating LAN Data Link users to agree (for any particular Protocol Type) whether the padded form or the basic form will be used.

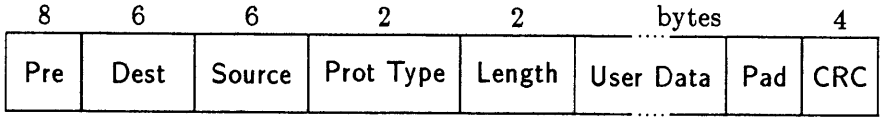


Figure 4.5: Ethernet frame format with padding

4.5 Bridges

Local Area Networks may be interconnected by LAN Bridges to create *Extended Local Area Networks*. A Bridge performs a Data Link layer relay function which is *transparent*

to the Data Link users, and independent of the protocols used above the Data Link layer. Bridges handle both ISO 8802-3 and Ethernet frames.

4.5.1 Bridge Services

The purpose of Bridges is to extend the LAN transparently, preserving the services available in the basic LAN (see Section 4.1) while increasing the number of stations allowed, the maximum distance between stations, and the total available bandwidth. The Bridge functions are layered on top of the LAN Data Link layer module. The constraints on LAN size (total number of stations, number of stations) are constraints of the layers below the Bridge functions; therefore they apply only to the individual LANs in a Extended LAN, not to the Extended LAN as a whole. Note that the Bridge uses the LAN Data Link layer module in such a way that it processes all frames, regardless of whether they are ISO 8802-3 or Ethernet format.

The bandwidth of a Extended LAN is greater than that of a LAN because Bridges only forward frames when needed. If the source and destination for a particular frame are on the same side of a Bridge, it will not forward the frame; other LANs in the Extended LAN can thus carry other frames at the same time.

4.5.2 Extended LAN Topology

Local Area Networks may be interconnected by Bridges with few topological constraints. Topologies with loops are permitted; in general the Extended LAN may have an arbitrary mesh topology. The constraints that do apply concern the total number of stations in the Extended LAN, and the number of Bridges in the path between any two stations (the network diameter). The total number of stations is limited by the capacity of the Forwarding database in the particular bridge implementation used; the network diameter is limited by the need to limit the total delay across the Extended LAN.

The advantage of this flexibility is that it allows topologies with redundancy. For example, a pair of LANs could be connected by two Bridges in parallel, or a number of LANs could be connected in a circular topology. In either case, the Extended LAN would remain intact even if one of the Bridges fails. More highly redundant topologies are also possible.

4.5.3 Bridge Algorithms

4.5.3.1 Spanning Tree Computation

As described above, the topology of an Extended LAN is relatively unconstrained and may include loops. Bridges compute a *spanning tree* for the Extended LAN to ensure frames will not loop continuously. Only Bridge ports that are in the spanning tree are used for frame forwarding.

The Bridges operate a protocol between them to exchange the information required to compute the spanning tree. Whenever the topology changes, the spanning tree computation

produces a new spanning tree. For example, if two bridges are in parallel, one will be in standby mode. Should the active bridge fail, the one in standby mode will take over.

4.5.3.2 Station Address Learning

Because Bridges forward packets using only ports that are in the spanning tree, there is only one path from a Bridge to any given destination station. This allows the Bridge to *learn* about stations by observing source addresses in received traffic, and forwarding based on the information learned.

Bridges maintain a local *forwarding database*, which contains station addresses and associated Bridge ports. Whenever an error free frame is received on a port, the Bridge enters the source address of the frame into the station address database, with the identification of the port on which the frame was received.

4.5.3.3 Forwarding algorithm

Whenever an error free frame is received on a Bridge port, the forwarding algorithm is used to determine how to process the frame. First, the Bridge checks whether the port on which the frame was received is in the spanning tree. If not, then this port is currently only a backup, and the frame is discarded. Next, the destination address is looked up in the forwarding database. If the port ID in the database is that of the port on which the frame was received, the Bridge simply discards the frame (since this indicates that the source and destination are "on the same side" of this Bridge). Otherwise, it forwards the frame on the indicated port.

If the destination address is not found in the forwarding database, this means the Bridge has no information about the location of the destination. In that case the frame is *flooded* along the spanning tree. The frame is forwarded on all ports in the spanning tree except the one on which the frame was received. The result of the flooding is that the frame is sent once on each LAN of the Extended LAN.

In short, Bridges *learn* from *source* addresses and *forward* based on *destination* addresses.

4.5.4 Bridge Management

Bridges are self-initializing and autoconfiguring. That is, they will operate correctly without requiring any management intervention.

Management facilities are provided to allow observation and diagnosis. They can also be used to make permanent entries in the forwarding database, for example to prevent certain traffic from being forwarded at all.

4.6 ISO 8802-3 and Ethernet Coexistence

DNA allows both ISO 8802-3 and Ethernet format frames to be used on the same LAN, and even on the same system. Each Data Link user specifies to the LAN Data Link module

which format is to be used (ISO 8802-3 LLC, ISO 8802-3 SNAP, basic Ethernet, Ethernet with padding).

The protocol frames can be distinguished by examining the two byte field following the Source Address. In ISO 8802-3 format, this is the length of the LLC data; in Ethernet format, the Protocol Type. The minimum size LLC frame has one byte DSAP, SSAP and Ctl fields and a null User Data field, and the maximum CSMA/CD frame size (not counting the preamble) is 1518 bytes; therefore the Length field in the ISO 8802-3 format will be in the range 3–1500.

When the CSMA/CD Data Link module receives a frame, it examines the two byte field following Source. If this field, interpreted as a two byte integer with the high order byte coming first, is in the range 3–1500, the frame is an ISO 8802-3 frame. Otherwise it is an Ethernet frame. As a result, the Ethernet Protocol Type codes in the range 00-03 to 05-DC are unavailable, since those values in the Prot Type field would be interpreted as ISO 8802-3 lengths. These values are therefore not assigned to any Ethernet Protocol Type.

DNA Phase V provides Ethernet support for compatibility with Phase IV and with other existing protocols that use Ethernet format frames. For new applications, the ISO 8802-3 format is preferred because it is an international standard, and also because it provides some LAN technology independence. An application using ISO 8802-3 could easily be migrated to other LANs in the ISO 8802 family, since the same set of services is available on all of them.

If an existing application that uses the LAN Data Link module needs to migrate to ISO 8802-3 format frames, some temporary duplication of function will be necessary during the migration to permit the application to interoperate with existing (Ethernet-only) versions. The new version would use ISO 8802-3 frame format as the default format, but would also listen for frames in Ethernet format. If a system using the older version attempts to communicate with the new version, the new version will notice that the request was made in Ethernet format and respond in the same format. For example, the DNA Phase V Network layer uses ISO Connectionless-mode Network protocol (ISO 8473) using ISO 8802-3 frame format, but also listens to Phase IV frames in Ethernet format. If it receives any, it records the fact that the sending system is a Phase IV system and sends appropriate responses in Phase IV format using Ethernet frames.

Chapter 5

Network Layer

The Network layer of DNA provides the ISO *Connectionless-mode Network Service* (CLNS) defined in ISO 8348 Addendum 1, using the protocol defined in ISO 8473. It also provides the ISO *Connection-mode Network Service* (CONS) defined in ISO 8348. The provision of CLNS is described below, while the provision of CONS is described in Chapter 9.

The Network layer accepts messages (NSDUs) from the local Transport layer modules via a local *Network Service Access Point* (NSAP) and delivers them to destination Transport layer modules via remote NSAPs. The particular destination NSAP is identified by means of its *NSAP address*. The information in the NSDU is transmitted through the network in one or more data *Network layer Protocol Data Units* (NPDUs). The Network layer is responsible for forwarding the data NPDUs, possibly through multiple intermediate systems, to the destination NSAP. In order to achieve this, it must be able to determine the best path (or paths) to each destination, based on the current topology of the network.

The network *topology* comprises a (possibly very large) number of systems interconnected by various *subnetworks* (such as DDCMP or HDLC point to point links, ISO 8802-3 LANs or X.25 packet switched networks). This topology will be constantly changing, both as a result of new systems and their connecting subnetworks being added, and the failure and subsequent repair of existing ones. In order to determine the best paths in the face of these perturbations, the DNA Network layer employs a distributed Routing Algorithm based upon the shared knowledge of all systems within a DNA *Routing Domain* (Routing Domains are described in Section 5.2.1). Each system is responsible for ascertaining the reachability and identity of its immediate *neighbors* (that is, those systems currently reachable by traversal of a single subnetwork). The information about the state of each system's links, together with a locally assigned *cost* for each link, is used to construct a *Link State PDU* (LSP) for that system. LSPs are propagated periodically (and whenever there is a change in any of the links to that system) throughout the routing domain.

As a result of receiving LSPs from all the other systems, each system is in possession of information about every other system's links. This information is used by each system to construct a representation of the current network topology and hence compute the best (that is, least cost) path to each particular destination.

5.1 Network Service Characteristics

The service provided by the Connectionless-mode Network Service has the following characteristics.

- It is a datagram service. Messages are delivered to their destination on a best-effort basis.
- Messages of up to 65535 bytes are accepted for transmission.
- There exists the possibility that the Network Service will:
  - Duplicate a message.
  - Deliver a sequence of messages to a given destination in a different order to that in which they were presented at the source.
  - Fail to deliver a message.
- There is an extremely low probability that the Network Service will deliver a message to the wrong destination or modify the data in a message.
- There is a variable delay time between the presentation of a message at the source and its subsequent delivery at the destination.

5.2 Network Layer Concepts

The following basic concepts are essential to an understanding of the operation of the Network layer.

5.2.1 Hierarchical Routing

A collection of End Systems, Intermediate Systems and Subnetworks operated by a single organization or administrative authority is known as an *Administrative Domain*. An Administrative Domain may be subdivided into a number of *Routing Domains*, some or all of which may be DNA Routing Domains.

A *Routing Domain* is a set of End Systems, Intermediate Systems and Subnetworks which operate according to the *same* routing procedures and which is wholly contained within a single Administrative Domain.

In order to support very large domains, possibly containing several hundred thousand systems, Routing Domains in Phase V DNA are themselves hierarchical. A large DNA Routing Domain may be partitioned by the network manager into sub-domains known as *areas*. Each system resides in exactly one area. Routing within an area is referred to as *Level 1 routing*. Routing between areas is referred to as *Level 2 routing*. (Routing between Routing Domains is performed by means of static tables as described in Section 5.2.5.) Level 2 routers keep track of the paths between areas. Level 1 routers keep track of the routing within their own area. For an NPDU destined to another area, a Level 1 router sends the NPDU to the nearest Level 2 router in its own area, regardless of the actual

destination area. The NPDU travels via Level 2 routing to the destination area, where it again travels via Level 1 routing to the destination system.

The division of a large DNA Routing Domain into separate areas improves network performance by drastically reducing the amount of routing overhead compared to a single area of the same size. It also allows the inter-area traffic to be confined to a particular set of systems and subnetworks.

5.2.2 System Types

The following types of system exist in DNA.

End Systems. These systems originate NPDUs for transmission to other systems and receive NPDUs from other systems, but do not relay NPDUs. In general End Systems only have one attached subnetwork. *Multilink End Systems* however, may have multiple attachments to one or more subnetworks (for example, to provide greater resilience to failures).

Intermediate Systems. These systems may originate and receive NPDUs from other systems, but also relay NPDUs from other source systems to other destination systems. In DNA, such systems are termed *routers*. There are two types of router.

Level 1 Routers. These systems route directly towards systems within their own area, and route towards a Level 2 router when the destination system is in a different area.

Level 2 Routers. These systems act as Level 1 routers within their own area, but in addition route to other areas. They route to other routing domains by means of static routing as described in Section 5.2.5.

5.2.3 Addressing

The structure of Network layer addresses within the global OSI environment is defined in ISO 8348 Addendum 2. This introduces the concept of *addressing domains*. At the highest level is the *global network addressing domain* which consists of all NSAP addresses in the OSI environment. This is subdivided into a number of sub-domains each of which is associated with an *addressing authority*. The authority may itself allocate addresses within its domain, or it may further subdivide its domain and assign an authority to each of its sub-domains. This process may be continued to an arbitrary extent, limited only by the maximum NSAP address length. The uniqueness of addresses within a particular hierarchical domain is ensured by the authority responsible for that domain.

The address format defined by ISO is illustrated in Figure 5.1. It divides an address into two major parts, the *Initial Domain Part* (IDP) and the *Domain Specific Part* (DSP). The IDP is further divided into an *Authority and Format Identifier* (AFI) and the *Initial Domain Identifier* (IDI).

The AFI specifies the format of the IDI and the authority responsible for allocating its values. ISO has specified a number of such authorities, among which are:

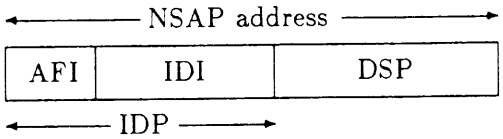


Figure 5.1: NSAP address structure

- X.121, the CCITT numbering plan for X.21 and X.25 networks.
- E.163, the CCITT numbering plan for the telephone network.
- E.164, the CCITT numbering plan to be used for ISDN networks.
- ISO Geographic. This is an ISO defined scheme which allows a country to assign addressing domains independently of public networks.
- ISO Non-geographic. This is an ISO scheme which allows international organizations to assign domains which do not imply the location of systems within the domain.

NSAP addresses in Phase V DNA may be constructed with any valid AFI and IDI.

To take a specific example, in the case of an AFI which specifies the X.121 numbering plan, the IDI is an X.121 address. This would typically, but not necessarily, be the address of the DTE by which a group of private subnetworks are attached to the public network. Although it is useful to think of a correspondence between an addressing domain and the physical topology of the network, it must be remembered that this is by no means mandatory. In this example the X.121 address need not correspond to *any* physical DTE at all. However it simplifies the routing between routing domains if such a correspondence can be achieved, since the DTE address to be called in order to reach a particular NSAP can then be derived from the NSAP address.

The format of the DSP is not subject to standardization and it is possible for different structures to be used in different routing domains, since it only need be interpreted within the defining domain. Within a DNA routing domain the DSP portion is structured as shown in Figure 5.2 in order to facilitate the operation of the routing functions.

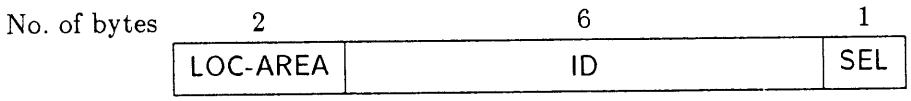


Figure 5.2: Structure of the DSP in DNA

LOC-AREA. A two-byte field which is unique within an IDP. The concatenation of IDP and LOC-AREA define the bounds of an *area* and is known as the *Area Address*. If

the Area Address of an NSAP address exactly matches the Area Address of a system, it is in that system's area, and is routed by Level 1 routing. Otherwise it is routed via Level 2 routing.

ID. A six-byte field which uniquely identifies the particular system within an area.

SEL. A one-byte field which acts as an NSAP selector for the module within a particular system which is to receive the data NPDU. This may be a Transport layer recipient or the Network Routing Module itself. The selector is always the last byte of the address.

In some circumstances, it may be desirable for an area (and hence the systems within it) to have more than one area address. For example, if the area has multiple points of attachment to public networks, it may be useful to have NSAP addresses which correspond to each one. However the routers making up an area must have at least one area address in common with each of their neighbors.

In routers, the area address(es) are set by Network Management. However, End Systems can autoconfigure their area addresses to be those of their neighboring router. Each DNA system has a unique 6 byte identification (known as the NodeID). This is used as the ID field of the NSAP address. The SEL field is specified by the Transport layer modules. End System NSAP addresses can therefore be completely autoconfigured without Network Management intervention.

5.2.4 Subnetwork Types

There are two generic types of subnetworks supported by the Network layer.

Broadcast Subnetwork. These are multiaccess subnetworks that support the capability of addressing a group of attached systems with a single NPDU, for instance ISO 8802-3 LANs (see Chapter 4).

General Topology Subnetworks. These are non-broadcast subnetworks, which may themselves be subdivided into two types:

Permanent Point-to-point Links. These are links that stay connected at all times (unless failed, or turned off by Network Management), such as leased lines or private links operating DDCMP or HDLC Data Link layer protocols.

Dynamically Established Data Links (DEDs). These are links over *connection oriented* facilities such as X.25 networks. They can be used in one of three ways:

Static Point-to-point (Static). The call is established upon Network Management action and cleared only on Network Management action (or failure).

Dynamic Connection Management (DCM). The call is established upon receipt of traffic, and brought down on timer expiration when idle.

Dynamically Assigned (DA). Uses dynamic connection management, but in addition, the address to which the call is to be established is determined dynamically from the destination NSAP address. Routing information is not exchanged over these links, but data traffic is forwarded according to statically entered routing information as described below.

5.2.5 Multiple Routing Domains – Static Routing

Systems within a DNA Routing Domain are able to communicate with OSI systems in other Routing Domains. These may be in other Administrative Domains, or separate Routing Domains within the same Administrative Domain. The latter may be desirable, in certain cases, for reasons of:

Cost. Only data traffic need be exchanged between the Routing Domains so no additional costs are incurred for routing control traffic.

Scale. With a two level hierarchy, there is a limit (albeit very large) to the size of Routing Domain the routing algorithm can handle. As long as each individual Routing Domain is reasonably sized, the total size of the Administrative Domain is for all practical purposes unbounded.

Robustness. Routing Domains are isolated from each other, and therefore protected from such conditions as faulty routers or excessively unstable topologies that might occur in a different Routing Domain.

Interoperability with other routing algorithms. The routing algorithm used in one Routing Domain does not need to be the same as the algorithm used within a different Routing Domain. Thus a DNA domain may communicate with systems in a non-DNA OSI domain.

In order to facilitate the construction of such multi-domain topologies, and permit data to be routed between domains, provision is made for the entering of *static* inter-domain routing information. This information is provided by a set of *Address Prefixes* entered by Network Management at the boundary Level 2 Router. The prefix indicates that any NSAP whose address matches the prefix may be reachable over the subnetwork with which the prefix is associated. The prefix also has associated with it the required subnetwork addressing information. In some cases, where the address of the subnetwork point of attachment has been used as the IDI in the NSAP addresses matched by the prefix, it is possible to derive the subnetwork address from the particular NSAP address. For example, an X.121 DTE address may be obtained from the IDI of the NSAP address. This permits a single static entry to be used to provide routing information for a number of different points of attachment.

The Address Prefixes are handled by the Level 2 routing algorithm in the same way as information about a Level 1 Area within the domain. NPDUs with a destination address matching a prefix present on any Level 2 router within the domain can therefore be relayed (using Level 2 routing) to that Level 2 router and delivered out of the domain. It is then

assumed that the routing functions of the other domain will then be able to deliver the NPDU to its destination.

5.3 Network Layer Functions

The Network layer Routing functions may be divided into two groups:

- Subnetwork Independent Functions
- Subnetwork Dependent Functions

5.3.1 Subnetwork Independent Functions

The Subnetwork Independent Functions are independent of the specific Data Link layer module operating below them. The following components may be identified.

Routing. The routing component determines NPDU paths. A path is the sequence of connected systems and links between a source End System and a destination End System. The routing component has the following specific functions:

- It extracts and interprets the routing PCI from received NPDUs.
- It performs NPDU forwarding based on the destination NSAP address.
- It manages the characteristics of the path. If a system or link fails on a path, it finds an alternative route.
- It interfaces with the Subnetwork Dependent Functions to receive reports concerning changes in the availability of neighboring systems.
- It returns error reports to the source where necessary.

Segmentation and reassembly. Where NPDUs are too large to be transmitted in a single Data Link layer SDU they are segmented. NPDUs are reassembled at the destination.

PDU lifetime control. This bounds the amount of time that a data NPDU can exist in the network, thus preventing NPDUs from circulating indefinitely. Each data NPDU has a lifetime field which is decremented every time the NPDU is forwarded. If the lifetime reaches zero, the NPDU is discarded and, if requested, an error report is returned to the source.

Congestion avoidance. This manages the buffer resources used at each intermediate system. When the average value of the queue of NPDUs for a link reaches a threshold, a *congestion experienced* bit is set in any NPDUs relayed over that link. This information is used by the Transport layer congestion avoidance functions described in Section 6.3.2.3. If the queue continues to grow, NPDUs are discarded to prevent further congestion. In addition the congestion avoidance function regulates the ratio of traffic being relayed by a system to traffic originated by that system.

5.3.2 Subnetwork Dependent Functions

The Subnetwork Dependent Functions mask the characteristics of the Data Link layer from the Subnetwork Independent Functions. They perform the following functions:

- Identifying whether the neighbor is a Phase IV or a Phase V system.
- Operation of the ISO *End System to Intermediate System Routing Protocol* (ISO 9542) on the particular subnetwork, in order to:
  - Ascertain the subnetwork addresses of operational neighboring systems, and the NSAP addresses present.
  - Ascertain the subnetwork address of operational routers.
  - Determine the Area Address to be used for the autoconfiguration of End System NSAP addresses.
- Operation of the requisite *Subnetwork Dependent Convergence Function* as defined in ISO 8473 Addendum 1, in order to perform:
  - Data Link initialization.
  - Hop by hop segmentation over subnetworks with small maximum SDU sizes.[1]
  - Call setup and tear-down on Dynamically Established Data Links.

5.4 Routing Operation

The Routing component contains three major processes.

- Decision
- Update
- Forwarding

5.4.1 The Decision Process

This process uses the database of link state information, derived from the LSPs received from other systems, to determine the least cost path(s) to each router in the routing domain. This information is entered into the *forwarding database*, from which the Forwarding Process can determine the proper next hop for each data NPDU.

[1]For example, the use of the M-bit procedures in X.25 as described in Section 9.3.1.3.

5.4.2 The Update Process

This process is responsible for generating and reliably propagating LSPs around the domain. LSPs are generated periodically and also when notified by the Subnetwork Dependent Functions that a change has occurred in the reachability of the system's neighbors. LSPs are propagated by *flooding*. Each new LSP is transmitted to all the router's neighbor routers except the neighbor router from which it received the LSP.

The Update process is responsible for determining, given a received LSP, whether that LSP represents new, old or duplicate information with respect to what is already stored in the database and acting accordingly.

- If the LSP is **new**, it is written into the database, acknowledged as described below, and forwarded on all links other than the one on which it was received.
- If the LSP is **old**, the stored, newer, LSP is transmitted on the link over which the older one was received.
- If the LSP is a **duplicate**, it is acknowledged, but is not forwarded.

Each LSP generated by a particular router is assigned a *sequence number*. An LSP with a preceding sequence number is defined to be *older*. Additionally, each LSP has an *age* field. When the age expires the LSP is deleted from the database.

The update process is responsible for ensuring that the latest LSPs reach every reachable router in the sub-domain. This is done by sending acknowledgments. On non-broadcast links these are sent whenever an LSP which is newer or a duplicate is received. Older LSPs are acknowledged by sending the newer LSP. On broadcast links a special *Sequence Numbers PDU* (SNP) is periodically multicast to all routers.

5.4.3 The Forwarding Process

This process inspects the destination NSAP address of a received data NPDU and determines, by reference to the forwarding database generated by the decision process, the correct link over which to forward the NPDU. It then queues it for transmission over that link. Where the forwarding database indicates that there are multiple equal cost paths to the destination, the forwarding process performs load splitting. Successive data NPDUs are transmitted on each of the indicated links in turn.

When the forwarding process is unable to deliver a data NPDU it is discarded. If the original NPDU so requested, an error NPDU is returned to the original source, specifying the system where the error occurred and the nature of the error.

When the forwarding process forwards a data NPDU onto the same subnetwork from which it was received, it additionally sends a *Redirect* NPDU (as defined in ISO 9542) to the originator of the data NPDU in order to inform it of the better route. Subsequent data NPDUs can then be sent directly to the system identified in the Redirect NPDU without further involvement of the router.

Chapter 6

Transport Layer

The Transport layer in DNA provides reliable transfer of data between Transport service users. It operates over the Network layer and can use either style of Network service: connectionless-mode (Chapter 5) or connection-mode (Chapter 9). The Transport layer employs end-to-end protocols which enhance the inherent characteristics of the underlying Network service.

6.1 Transport Layer Functions

The DNA Transport layer performs the functions described below. DNA can operate two different Transport layer protocols (Section 6.2). The exact functions performed will depend upon the protocol being used.

Construction and destruction of Transport Connections. A Transport Connection is a virtual connection at the Transport layer between two Transport service users, either on separate systems or the same system in a network. A pair of Transport service users may have more than one Transport Connection between them. Each Transport Connection in a network exists independently of any other. A Transport Connection must be established before two Transport service users can exchange data. The process of connection establishment also permits the communicating peers to negotiate mutually acceptable characteristics for the connection.

Data transfer using flow control. Transport provides a full duplex data path between a pair of communicating Transport service users. It transfers data from transmit buffers in one system to receive buffers in another. Flow control mechanisms are employed to balance the relative speeds of the transmitter and receiver. Transport recovers from message loss or duplication occuring in lower layers and ensures that data is passed to the receiving Transport service user in the same order in which it was transmitted.

Congestion avoidance. Congestion occurs when a network, or part of a network, is overloaded and has insufficient communications resources for the volume of traffic.

Congestion can result in messages being discarded by the Network layer. Recovery from lost messages is expensive, wasting system and network resources as well as reducing the users' throughput and increasing the delay.

The Transport layer employs congestion avoidance algorithms to adjust the load such that the network operates at a point where message loss due to congestion is unlikely. It uses information provided by the Network layer and attempts to keep the users' throughput to a maximum while sharing network resources fairly. Additionally, it employs recovery mechanisms to respond to transient overloads which cause message loss.

Segmentation and reassembly. Transport permits its users to transmit extremely large messages. If the underlying Network service is unable to accept a single message of the desired size, Transport performs segmentation of the user message into smaller messages for transmission and the reconstruction of the complete message from the received segments.

Multiplexing. When operating over a connection-mode Network service, Transport permits more than one Transport connection to be multiplexed onto a single Network connection. Each Transport message carries a connection identifier, assigned during connection establishment, which permits the receiving Transport to determine the connection with which the message should be associated.

Concatenation. Transport may concatenate certain messages from the same Transport connection and pass them to the Network layer as a single message for transmission. This is sometimes referred to as *piggybacking* and reduces the number of Network layer messages transmitted.

Error detection and recovery. The Transport Layer provides detection and recovery from loss, duplication, corruption and misordering occuring in lower layers. It employs message numbering and a basic acknowledgment mechanism to ensure that messages are delivered, and an optional checksum capability to detect message corruption.

Management. The Transport layer provides facilities to monitor and control the operation of the Transport layer itself and of individual Transport Connections.

6.2 Transport Layer Protocols

Two different protocols can be operated by the DNA Transport Layer. The first of these is the OSI Transport protocol specified in International Standard ISO 8073. This standard defines five *classes* of protocol, numbered 0 to 4. DNA provides classes 4, 2 and 0. The OSI Transport protocol is being introduced into DNA for the first time in Phase V. As well as being available for communication between Phase V DNA systems, it also permits Transport layer communication with non-DNA systems which implement this protocol.

The second protocol which may be operated by the DNA Transport layer is called the Network Services Protocol (NSP). NSP was designed specifically for DNA and has been a part of DNA since its inception. It offers communication between Phase V DNA systems and provides backwards compatibility with Phase IV DNA systems.

The DNA Transport layer integrates these protocols and offers a single, consistent service interface to the Transport service user. The choice of protocol for a particular instance of communication is made during connection establishment. In practice, there exist certain differences between the protocols which must be reflected in the Transport service. Most such differences are minor and the Transport service user may use the protocols interchangeably.

The DNA Transport layer operates over the DNA Network layer and can use both types of service provided by the Network layer: connectionless-mode (CLNS) and connection-mode (CONS). Certain of the Transport protocols may only be used with a particular type of Network service. Table 6.1 shows the combinations of Transport protocol and Network service which are supported.

	NSP	*OSI Class 0*	*OSI Class 2*	*OSI Class 4*
CLNS	Yes	No	No	Yes
CONS	No	Yes	Yes	Yes

Table 6.1: Transport Protocols and Network Services

6.3 OSI Transport Protocol

The OSI Transport protocol is defined by International Standard ISO 8073. This standard defines five *classes* of protocol, the following three of which are provided by DNA.

class 0 Minimal features
class 2 Multiplexing class
class 4 Error detection and recovery class

These classes are actually distinct protocols which share a common message structure and employ a common connection establishment procedure. During connection establishment the class of protocol to be operated is dynamically agreed by the participants.

The class 4 protocol offers the richest set of capabilities and can operate over both the connectionless-mode and connection-mode styles of Network service. This protocol can do everything that the other classes can do and is suitable for all purposes. It is expected to be the most widely used.

Classes 2 and 0 are simpler protocols and have successively reduced capabilities. In particular, they can only operate over the CONS. Use of the class 0 protocol is required

by the CCITT X.400 messaging standards when communicating with an Administration Domain. Provision of the class 2 protocol offers a further choice which will increase the level of interoperability between Phase V DNA systems and non-DNA systems.

The following sections describe in more detail some of the major features of the OSI Transport protocols and how they are operated by DNA.

6.3.1 Connection Establishment

Connection establishment is an operation during which the class of protocol to be operated and other characteristics of the connection are agreed between the participants. A connection is initiated by the transmission of a *connection request* message which identifies the target Transport service user at the remote system. It also indicates a preferred class of protocol and any acceptable alternatives, and various characteristics of the desired connection; for example, the maximum message size and whether subsequent messages should carry checksums to guard against corruption. The received connection request is accepted by the transmission of a *connection confirm* message. This message carries the choice of protocol to be used, and the actual characteristics for the connection negotiated according to rules defined by ISO 8073.

6.3.2 Data Transfer

Once connection establishment has been successfully performed, the Transport Connection enters the data transfer phase. The communicating service users are then provided with a full duplex path for the exchange of data.

6.3.2.1 Flow Control

Classes 4 and 2 operate data message numbering for reliable, sequenced delivery. A sequence number from a circular number space is assigned to each data message transmitted. As data messages arrive at their destination, the receiver returns an acknowledgment message to the transmitter indicating the message number up to which data messages have been successfully received.

These protocols implement flow control according to the notion of *credit*. This means that the receiver of data tells the transmitter how many messages it is prepared to receive at a given point. Flow control operates in conjunction with the message numbering space.

The class 0 protocol has no explicit flow control. It does not number data messages and does not send acknowledgment messages. Instead, flow control in the CONS is used to match the speeds of transmitter and receiver.

6.3.2.2 Data Retransmission

The class 4 Transport protocol employs a *retransmission timer* to detect message loss. The timer is started when the message is first transmitted and stopped when the message's acknowledgment is received. If the timer expires, the message is assumed to have been lost and is retransmitted.

The interval of the retransmission timer must be chosen carefully such that it is short enough to ensure that loss of a data message is detected quickly, but not so short that messages get retransmitted unnecessarily. DNA operates an adaptive algorithm which maintains an average estimate of the round-trip delay on each Transport Connection. The value used for the retransmission timer is then a simple function of the estimated round-trip delay. The estimate is updated by continually sampling the round-trip delay and incorporating new readings with the current estimate. Consequently, the value used for the retransmission timer varies in sympathy with changes in the round-trip delay.

6.3.2.3 Congestion Avoidance

In DNA, operation of the class 4 protocol over the CLNS employs two algorithms which adapt to the changing load on the network and prevent congestion.

The first operates in conjunction with the Network service and employs its *congestion experienced* indication (Section 5.3.1). If, during a sample period, the number of received messages which encountered congestion exceeds an architecturally defined threshold, the flow control window size is reduced. This, in turn, reduces the number of messages which may enter the network and so reduces the load. If the threshold is not exceeded, the window size may be increased.

The second algorithm is invoked whenever message loss occurs on a connection. This is assumed to indicate serious congestion and the transmitter then restricts the speed at which it injects further data messages into the network. It operates a local credit window which is smaller than the full window which it has been allocated by the receiver. This local window is initially assigned a value of one, permitting the message which timed-out to be retransmitted. Its value is then increased by one each time the number of data messages for which acknowledgments have been received since the last change becomes greater than the current value of the local credit window. This mode of operation is continued until the local credit window grows to the value of the actual credit window assigned by the receiver. Normal operation is then resumed.

6.3.2.4 Expedited Data

Protocol classes 4 and 2 provide an additional data transfer service called *expedited data.*[1] This service permits a single, short data message to be transmitted which will bypass any blockage due to normal data flow control. The class 0 protocol does not provide this service.

6.3.2.5 Reassignment After Failure

When the class 4 protocol is operated over the CONS, DNA attempts to maintain the Transport Connection even if the underlying Network Connection fails. This is achieved either by reassigning the Transport Connection to some other existing Network Connection, or by establishing a new Network Connection for the purpose. The reassignment operation is transparent to the Transport service user.

[1] Despite what its name may imply, expedited data is not necessarily transferred any faster than normal data.

6.3.3 Disconnection

A Transport Connection may be disconnected at any time, either by one of the Transport service users, or by one of the peer Transport modules. In protocol classes 4 and 2, a *disconnect request* message is transmitted and the receiver responds with a *disconnect confirm* message. In protocol class 0, the Transport Connection is terminated simply by disconnecting the underlying Network Connection.

6.4 NSP

The second Transport layer protocol provided by DNA is called the Network Services Protocol (NSP). This protocol has many similarities with the class 4 OSI Transport protocol and provides an almost identical service. This section describes certain features of NSP operation in more detail.

6.4.1 Connection Establishment

NSP establishes, maintains, and destroys Transport Connections by exchanging control messages with a peer NSP module. An established connection is made up of two separate data subchannels, each carrying messages in both directions:

Normal data subchannel. This subchannel carries Data-Segment messages between two NSP modules.

Other-data subchannel. This subchannel carries Expedited Data messages and Flow Control messages.

6.4.2 Data Transfer

6.4.2.1 Segmentation and Reassembly of Data

The Network layer limits the amount of user data that NSP can send in one message. Taking normal data from its client's buffers, NSP breaks it up, if necessary, into segments. Each segment is numbered and sent along with its number and other control information in a Data-Segment message to the receiving NSP module. The receiving NSP module uses the sequence numbers to reassemble the data segments in correct sequence in the receiving user buffers. NSP segments normal data only. Expedited data messages do not require segmenting since their limited size guarantees that they will always fit in a single Network layer message.

6.4.2.2 Flow Control

NSP's flow control mechanisms ensure that data is not lost for lack of receiver buffering capability and that deadlocks do not occur. Both normal and other-data subchannels are flow-controlled. When a Transport Connection is formed, each NSP module informs the other of the way it wants to control the flow of normal data as a data receiver. The receiving NSP chooses either of two types of normal data flow control:

On/Off only. The receiver tells the transmitter to stop or start sending data.

Segment with On/Off. The receiver sends a request count of the number of segments it can accept. In addition, the receiver can always tell the transmitter either to stop sending data unconditionally or to start sending data under the normal request count conditions.

The receiver also controls other-data flow with an Other-Data request count. The Data Request message, Expedited Data message, Other-Data Request message, and the Other-Data-Ack message may contain an acknowledgment for a Data-Segment message and/or an Other-Data message. This ability, to combine acknowledgments with data and control messages, reduces the number of NSP messages required per user message.

6.4.2.3 Data Retransmission

The NSP modules at each end of a Transport Connection positively acknowledge received data. If the transmitting NSP fails to receive a positive acknowledgment during a timeout interval, it retransmits the data. As for the class 4 OSI Transport protocol, the timeout interval is adjusted dynamically based upon the round-trip delay observed by the transmitter.

6.4.2.4 Congestion Avoidance

While the flow control mechanisms protect against lack of receiver buffering, they do not deal with insufficient resources in other parts of the network. NSP employs congestion avoidance mechanisms, similar to those of the OSI Transport protocol module, to adapt to the changing load on the network.

The primary difference between the NSP and OSI Transport congestion avoidance mechanisms is that the congestion information reported by the Network layer is relayed to the data-transmitting NSP. Based upon this congestion signal, the data-transmitting NSP adjusts the maximum number of sent but unacknowledged data segments it is allowed to have outstanding.

6.4.3 Disconnection

A connection may be disconnected at any time, either by one of the communicating Transport service users, or by one of the peer NSP modules.

Chapter 7

Session Control

The Session Control module, a client of the Transport layer, provides functions for system-dependent process-to-process communication, name to address mapping, and protocol selection. These functions bridge the gap between the Transport layer services and the functions required by applications running under an operating system.

7.1 Functional Description

Session Control functions include:

- Matching incoming connect requests to the appropriate users of Session Control.
- Managing Transport Connections on behalf of users.
- Enforcing access control policies to restrict communication between users and between Session Control modules.
- Mapping from a DNA Naming Service object name to protocols and addresses.
- Selecting communications protocols supported by both the local and a remote end system.
- Maintaining in the Namespace the protocol and address information corresponding to local objects (see Chapter 8 on the Naming Service for the definition of a Namespace and an object).

7.2 Session Control Functional Components

The Session Control module enhances the Transport service available to a network application. The purpose of Session Control is to form a bridge between the applications requiring Transport service and the Transport module which actually creates, maintains, and destroys Transport Connections.

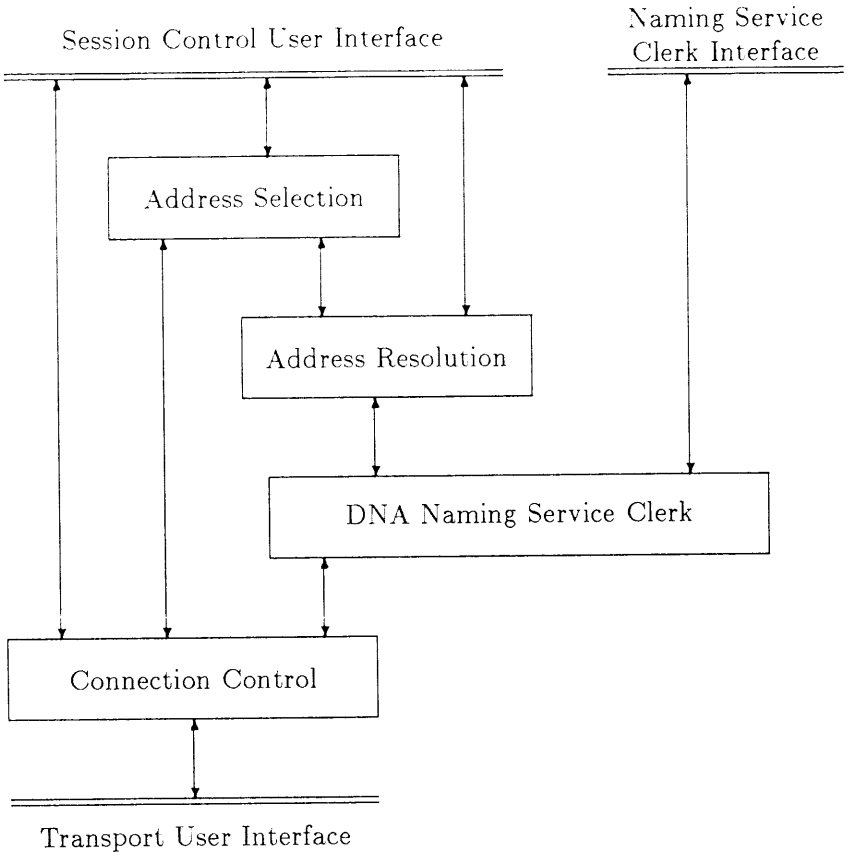


Figure 7.1: Session Control Components

Session Control is composed of three functional components, *Connection Control, Address Resolution* and *Address Selection.* The relationship among these functional components and between these components and the DNA Naming Service (see Chapter 8) is shown in Figure 7.1. The vertical arrows indicate the flows of information with components below providing services to those above.

7.3 Connection Control

Connection Control is concerned with the system-dependent functions related to creating, maintaining, and destroying Transport Connections. Connection Control processes requests for Transport Connections using the protocols and addresses passed by the application. It also enforces the locally defined access control policies.

The functions provided by Connection Control include:

- Requesting an outbound Transport Connection to an application based upon the

addresses and protocols specified in the request.

- Receiving an incoming Transport Connection request from the Transport layer.
- Validating access control information.
- Sending and receiving data.
- Monitoring a Transport Connection.
- Terminating a Transport Connection.

Many of these functions are similar to those that Connection Control requests from Transport. In response to requests from users, Connection Control makes corresponding requests to Transport. The value added functions that Connection Control provides over the basic Transport Connection service are described in the following sections.

7.3.1 Requesting a Connection by Destination Address

A connect request in which an application specifies all the protocols and addressing information is handled exclusively by the Connection Control component.

Connection Control performs the following functions:

- Formats connect data.
- Issues a connect request to the Transport layer protocol module specified by the application.
- Starts an outgoing connection timer, if the user so requests. Expiration of the timer prior to an acceptance or rejection from the Transport layer causes Session Control to disconnect the Transport Connection for the source application.

7.3.2 Receiving a Connect Request

Upon receipt of an incoming connect request from the Transport layer, Connection Control performs the following tasks:

- Parses connect data to obtain such information as source and destination application descriptors and access control information.
- Validates any access control information.
- Identifies, creates, or activates a destination application context.
- Delivers the incoming connect request to the application.
- Starts an incoming timer when the connect request is delivered. Expiration of the timer before the application accepts the connect request causes Session Control to issue a reject to the Transport layer.

7.3.2.1 Validating Access Control Information

Connection Control uses access control information included in an incoming connect request to perform system-dependent validation functions. The information provided may be either an explicit access control string (including a password) or a request to invoke a *proxy* on behalf of the requesting user.

Proxy mapping is a mechanism by which a user on one system in a network can be given access to accounts on another system in a network without knowing the access control information of the target accounts. This is accomplished by setting up a mapping on the target system between the remote user and local, or proxy accounts. Upon receipt of a connect request containing a (possibly null) proxy account name and an indication that a proxy is desired, the appropriate proxy account is selected.

7.3.2.2 Identifying Applications

Connection Control executes a system-dependent algorithm to ascertain if an existing application corresponds to the destination application specified in an incoming connect request. Connection Control may have an operating system interface to create a new user context in which to run the application. Once the identity of the application is determined, the connect request is passed to this context.

7.3.3 Sending and Receiving Data

This is a system-dependent function that handles application requests to send and receive data. These functions are passed directly to the Transport layer.

7.3.4 Monitoring a Transport Connection

If requested by the user, Connection Control will:

- Detect probable network disconnection between the systems as reported by the Transport layer.
- Detect a failure to respond promptly to a connection request.

If either of these events is detected, then Session Control terminates the Transport Connection.

7.3.5 Disconnecting or Aborting a Transport Connection

If the application requests disconnection, Session Control will wait until all previously transmitted data has been acknowledged by the remote transport module before issuing a disconnect request to Transport.

Session Control immediately issues a disconnect request to the Transport layer in response to an application abort request. Previously transmitted but unacknowledged data may not be delivered to the remote application.

Notification of a Transport layer disconnect or abort initiated by the remote Transport is passed directly to the application along with any disconnect data.

7.4 Address Resolution

Address Resolution maps from a Naming Service object name to the identifiers of protocols and corresponding addresses which are mutually supported by the local system and the remote system(s) on which the named object resides. A *Protocol Identifier* is a character string name for a protocol.

Accompanying these protocol identifiers is the address and protocol specific information necessary for protocol entities to exchange data. From this information, a sequence of protocols and associated local and remote addresses may be selected to attempt communication with the named object.

Previous versions of Session Control required each application to maintain, via network management, a static node name to address mapping table. The size of the table, and consequently the number of node names known to Session Control, was limited by local storage considerations. Furthermore, the information in this mapping table was sometimes out of date resulting in incorrectly addressed messages.

By relying on the Naming Service to store the mapping from names to addresses, the number of destinations accessible to any given system is no longer constrained by the quantity of local storage available. Because the Address Resolution component automatically maintains this name to address mapping in the Naming Service for local objects, the data is more up to date.

The functions provided by Address Resolution include:

- Maintaining in the Namespace the protocol and address information corresponding to local objects.
- Obtaining information about an object from the Naming Service for the purpose of establishing the sequences of protocols and address information for communication with the object.
- Caching protocol and address information to enhance performance.

7.4.1 Towers

In previous versions of DNA, there was no choice of the protocol operated at each layer below the application, and each system had only a single Network layer address. In Phase V, a system may support multiple Transport protocols (for example, NSP and OSI Transport) and may have multiple Network layer addresses.

In order for applications to communicate, they must agree upon the protocols that both will employ and the operational parameters of these protocols. In addition, these users must possess address information that indicates to each layer of protocol where to deliver data. This information is encoded in a *Tower*.

A *Tower* is a *Protocol Sequence* along with associated address and protocol specific information. A *Protocol Sequence* is an ordered list of protocol identifiers. Associated with each protocol identifier in a Tower is a component of the address and other protocol specific information. The address information indicates the access point through which this layer

Protocol Identifiers	*Addressing Information*
⋮	⋮
Layer $n+1$ protocol identifier	Layer $n-1$ parameters and addressing to reach $n-2$ layer module
Layer n protocol identifier	Layer n parameters and addressing to reach $n+1$ layer module
Layer $n-1$ protocol identifier	Layer $n-1$ parameters and addressing to reach n layer module
⋮	⋮

Figure 7.2: Structure of a *Tower*

provides service to the next higher-layer protocol in the sequence. Other protocol-specific information may be included in this field. Figure 7.2 illustrates the structure of a Tower. Typically, a tower will extend from the DNA Application layer to the Network layer and multiple Towers will be associated with an object.

7.4.2 Establishing Protocol Sequences for Communication

The Address Resolution component returns, in response to a client request, the protocol sequences supported by both the local system and a remote system. A client of the Address Resolution service passes both the name of a remote object and a local set of protocol sequence for all layers above Session Control. The protocol sequences passed by the client are combined with those below Session Control to determine the set of local protocol towers of interest. The local protocol and address information supporting Session Control from below is supplied by the Transport layer.

The Address Resolution component will then call the Naming Service Clerk to obtain the set of Towers associated with the object name. The protocol sequences from the Towers corresponding to the named object are compared with the local set of Towers to find the intersection of matching sequences. A pair of protocol sequences are said to match if the protocol identifiers in one member of the pair are identical to, are in the same order as, and map one to one with those in the other member. Each matching sequence found is returned to the requester along with the address information from both the local and remote Towers.

A user may then select from this set the protocols and addresses to attempt communication with the object.

To improve performance, protocol sequence and address pairs are cached for future use. An application may request that the information cached about a name be discarded and new protocol sequences and address pairs generated. Typically, this would be requested if a connection attempt fails.

7.4.3 Maintaining the Towers in the Namespace

Because the Network layer addresses are autoconfigured, the addresses associated with protocols supporting Session Control may change with time. Session Control includes a mechanism by which it can be asked to maintain the protocol and address information (that is, the set of Towers) in the namespace for a set of object names.

Using information about the higher layers which is passed to Session Control from above, and information about the lower layers requested by Session Control from each underlying Transport protocol module, Session Control updates the protocol and address information stored in the namespace for each such object by making requests to the Naming Service Clerk.

7.5 Address Selection

Address Selection allows an application to initiate connections based upon the name of an object. This relieves the user of both selecting protocols below Session Control and locating address information. This component relies upon the services of Address Resolution to perform name to protocol sequence and address mapping, as well as Connection Control to establish Transport Connections.

Address Selection attempts to find a set of protocols mutually supported by the local and remote systems. A protocol sequence and associated address information are selected from this set based upon local system policy. The selected protocols and address information are then used for communication with the remote object. In the face of failures, Address Selection will retry Transport Connection establishment with another member from the set of mutually supported protocols and address information.

7.5.1 Requesting a Transport Connection by Destination Name

An application may request a Transport Connection to a named object either by Node Name or by any object name.

7.5.1.1 Connecting by Node Name

For Phase IV style naming, an application may present Session Control with the destination Node Name and application address information. For compatibility with existing applications, Address Selection accepts a six-character node alias which it maps to the Node Name. Through the services of the Address Resolution component, the transport and network layer protocols and associated addressing information will be generated.

7.5.1.2 Connecting by Object Name

The user may present Session Control with an object name. Through the services of the Address Resolution component, both the higher layer addressing information as well as the transport and network layer protocols and addresses will be generated.

7.5.1.3 Ordering the Protocol Sequences

As described above, the object name passed by an application in a connect request is mapped to a set of protocol sequences and addresses via the Address Resolution services. The Address Selection component then orders the members of this set in a system-specific fashion. The services of the Communication Control component are subsequently invoked to attempt Transport Connection establishment using the the first protocol sequence and addresses from the ordered list.

7.5.1.4 Retry on Failure

If the initial attempt to establish a Transport Connection fails, the next protocol sequence and associated address information are selected from the ordered set. Transport Connection establishment is again requested with these new parameters. The Address Selection component will continue to step through this ordered set and retry connection establishment upon failure until either the set is exhausted or the reason for failure indicates that further attempts would be futile.

Chapter 8

Naming Service

The Naming Service is a new component of DNA for Phase V. It is closely coupled with the Session Control Layer, but its functions are directly available to network applications as well. The fundamental purpose of the Naming Service is to enhance the logical organization of large networks by allowing the *names* of network applications, systems, and other network-accessible objects to be independent of their physical location (address) in the network. Just as the Network layer provides the "glue" that organizes the physical structure of the network from its topology of lines and systems, the Naming Service provides the glue that organizes the logical structure of the network from the available application-level resources.

The Naming Service associates strings of characters, known as *primitive names*, with a set of *attributes* of the named object. For example, one of the named objects processed by the Naming Service is the *node* object.[1] The name of each system in the network is stored in the Naming Service. Along with the name an attribute is stored which contains the system's network address. This allows applications to refer to systems by a consistent, global name which is guaranteed to be unique. Storing system names in the Naming Service also eliminates the need for a local *node database*, which performed a similar function in DNA Phase IV.

8.1 Naming Service Concepts

Each name stored in the Naming Service refers to a single, unique object. Names for objects are recorded in *Directories*, which themselves have primitive names. A directory contains three types of entries, *object entries*, *child directory entries*, and *soft links*. An object entry consists of the object's name and a set of attributes for the object, most prominent of which is the *Network Address* where the object currently resides. Child Directory entries link the directories together into a rooted tree, in which there is a single path from the *Root Directory* through a set of child directories, to the desired named object. A soft link is a form of "alias" or "indirect pointer" which allows a single entry to be reachable via more than one name. Soft links both provide for user flexibility by allowing a namespace

[1] See Section 10.1.1 for a description of how this is used by network managment.

to be viewed as a directed graph rather than as a pure tree, and provide a graceful way to reorganize a namespace.

A tree of directories, starting at a root, is called a *namespace*. Multiple namespaces may coexist on the same network, but they are completely disjoint.[2] A namespace is uniquely identified by a *Namespace Unique Identifier (NSUID)*, assigned when the namespace is created. Namespaces do not have names as far as the Naming Service is concerned, but are assigned a *Nickname* so clients do not have to directly process the NSUID.

8.2 The Semantics and Syntax of Names

8.2.1 The Semantics of Names

A Name is a complete path specification from the root directory of a namespace to the particular entry in the directory of interest. Sometimes, however, it is necessary to refer to the part of the name which specifies a particular object entry or child directory entry in a directory as opposed to the entire Name. In these cases the terms *simple name* and *full name* respectively are used. Every object entry and every soft link has a simple name. Every directory has a simple name with the exception of the root directory of a namespace, which has no name. A simple name denotes exactly one of either an object entry, a child directory entry, or a soft link in a given directory.

All Names processed by the Naming Service through the client interface are full names, which have the property of *absoluteness* or *referential transparency*. This means that a full name refers to the same entry no matter which client provided the name, and that these full names may be freely passed outside of the Naming Service (for example, on pieces of paper) from one client to another without the possibility of confusion. Full names can be quite cumbersome to use, however, since they can get rather long and contain the NSUID of the namespace. While the Naming Service does not specify any mechanisms to alleviate this problem, Digital operating systems on which the Naming Service runs contain some native mechanisms for *nicknames*, which are a purely local shorthand for all, or part of a full name (for example, VMS logical names).

8.2.2 The Syntax of Names

There are two syntaxes for Naming Service names, the *external name* and the *internal name*. The external name is the syntax used for the human-readable form of a name. The internal name is the syntax of the name as passed across the client interface with the Naming Service. The external name is designed for readability while the internal name is designed to be convenient to encode in programs, protocols, and databases.

An external name consists of two parts: a namespace nickname and a sequence of simple names. The namespace nickname is translated locally by the Naming Service into the equivalent NSUID. If a namespace nickname is omitted from an external name, the

[2] In general it is undesirable to have multiple namespaces on a network because they are significantly harder to use and manage than a single namespace. The existence of multiple namespaces is a temporary phenomenon which arises when previously disjoint networks are merged into a single network.

Naming Service chooses a default namespace based on a local, implementation-dependent algorithm.

The namespace nickname and each simple name in the sequence that represents the full name is an ordered list of strings of letters, digits and certain punctuation characters from the Digital *multinational character set*. The case of each string is preserved by the Naming Service: a name registered with a mix of upper and lower case characters will appear when later retrieved exactly as entered. Lookups however, are case-insensitive. The strings "AbCöE", "ABCÖE" and "abcöe" all represent the same simple name.

The simple names in the sequence are delimited by a period ".". If a client desires to use the "." (or any other disallowed punctuation character) as part of a simple name the name must be enclosed in double-quotes. Figure 8.1 contains some example full names.

```
Parts.widgets.left-handed.SMOKESHIFTER
Government:Treasury.Bills.CurrentSeries
DEC:.Engineering.Networks.Arch.Specs
ULTRIX.Sources."OSITransport.c"
```

Figure 8.1: Examples of External Names

8.3 Contents of Namespace Entries

All Naming Service entries (objects, directories, soft links, and child directories) consist of a set of *attributes* and their associated value or values. Attributes are of two types. When an attribute may take on more than one value, it is called an *attribute set*; if it may have only one value at a time it is called a *single valued attribute*. An attribute set operates as a "dictionary" which allows the following operations:

Redundant Insert: A new value may be inserted in the set. If the new value matches a value already present, a separate entry is not made. The client has no control over the ordering of the members of the set.

Redundant Delete: An existing value may be removed from the set. If the value is not present, no error is returned.

Full Lookup: The entire contents of the set may be returned

Attributes fall into two general categories, *global attributes* and *class specific attributes*. Global attributes are those whose meaning is the same for *all* entries stored by the Naming Service. Class specific attributes are attributes whose definition is dependent on the value of one of the global attributes: the *class attribute*.

8.3.1 Global Attributes

Table 8.1 lists some of the global attributes defined by the Naming Service along with a description of the function of the attribute and an indication of the kind of Naming Service entry with which it is used.

8.3.2 Predefined Object Classes

Object entries, in addition to their use by clients directly, are also used by the Naming Service itself to provide certain functions, such as the ability to give clearinghouses human-sensible names, and to provide the notion of a group. Table 8.2 lists some object classes that are predefined by the Naming Service.

8.4 Operational Concepts and Terminology

A namespace is stored in a *partitioned, partially replicated* database. The database is partitioned because parts of the namespace are stored in different locations (that is, on different systems). The database is partially replicated because part of the namespace may be simultaneously stored in multiple locations. The unit of both partitioning and replication is the directory. A collection of (copies of) directories stored on a particular system is called a *Clearinghouse*. The partitioning is accomplished by controlling which directories are stored in which clearinghouses. The replication is accomplished by storing a directory in more than one clearinghouse.

Clearinghouses have names so that clients may conveniently refer to them, and so that the Naming Service can find a clearinghouse by looking up its name in the namespace. The naming of clearinghouses follows a set of strict rules which ensure that a name lookup cannot fail because the clearinghouse in which the directory storing a particular entry cannot be found.

Clearinghouses are either "active" or "inactive". When a clearinghouse is active at a given system, that system is acting as a *nameserver*. A nameserver may be controlling more than one clearinghouse simultaneously, especially when a clearinghouse has been moved after the failure of a nameserver.

A copy of a directory stored in a particular clearinghouse is called a *replica*. In order to simplify the algorithms for general namespace maintenance, one of the replicas of a directory is designated to be the *master replica* for that directory. Master replicas perform certain overhead functions that would be redundant if carried out independently by every replica of a directory. A second kind of replica is the *secondary replica*. Creation of new objects, directories, and soft links may be done at either a master or secondary replica. Updates to existing entries and deletion of entries may also be done at either kind of replica. A third kind of replica, the *read-only replica*, responds only to lookup requests and is not permitted to perform creations, updates, or deletions on behalf of clients.

The Naming Service maintains a distributed database on behalf of its clients. This database does not have the usual characteristics of a distributed database since it provides

Attribute	Entries	Description
UID	All	A *Unique Identifier* for the entry. This is a space and time unique identifier for the entry which is assigned when the entry is made and is never changed.
UTS	All	An *Update Timestamp* indicating the Timestamp of the most recent update to an entry.
ACS	All	The Access Control Set for the entry. Each value is an Authorization Entry (AE).
Class	Object	An Attribute used to classify object entries according to the type of object being named. This attribute allows clients to correctly interpret the contents of the class specific attributes of an entry.
Address	Object	Contains a set of Network Addresses where the object currently resides.
Replicas	Directory	Each member of this attribute set identifies one of the clearinghouses which stores a replica of this directory.
AllUpTo	Directory	This attribute contains a bound on how out of date various replicas of this directory are. All replicas of a directory are guaranteed to have received all updates whose timestamps are less than the value of AllUpTo.
Convergence	Directory	This attributes specifies how persistent a directory should be in trying to keep its replicas up to date.
ParentPointer	Directory	This attribute links a child directory to its parent, allowing the Naming Service to work up the tree as well as down. It is used to ensure that parent and child directories always point to each other during normal operation, and is used during directory creation to allow a child directory to link itself into the tree by creating a child directory entry in the parent directory.
LinkTarget	Soft Link	Contains the full name of the entry the soft link points to.

Table 8.1: Naming Service Global Attributes

Object Class	Description
Group	Defines a group. The members of the group may be either principals (names which identify clients of the name service) or other objects of class **Group**. A client interface operation, **TestGroup**, is provided to test for Group Membership. This operation is capable of recursively examining groups within groups, and can detect loops in group membership.
Clearinghouse	The Clearinghouse Object is used internally by the Naming Service for locating Clearinghouses.

Table 8.2: Naming Service Predefined Object Classes

very "loose" consistency guarantees in order to allow high levels of partitioning and replication. A client may get different answers depending which replica of a directory is queried if updates are still being propagated through the network.

Updates to a namespace are timestamped and applied such that the update with the latest timestamp wins. The updating algorithms are designed such that all updates are *idempotent* (multiple applications of an update to the database have the same effect as a single application of the update) and *commutative* (updates may be applied in any order with identical results).

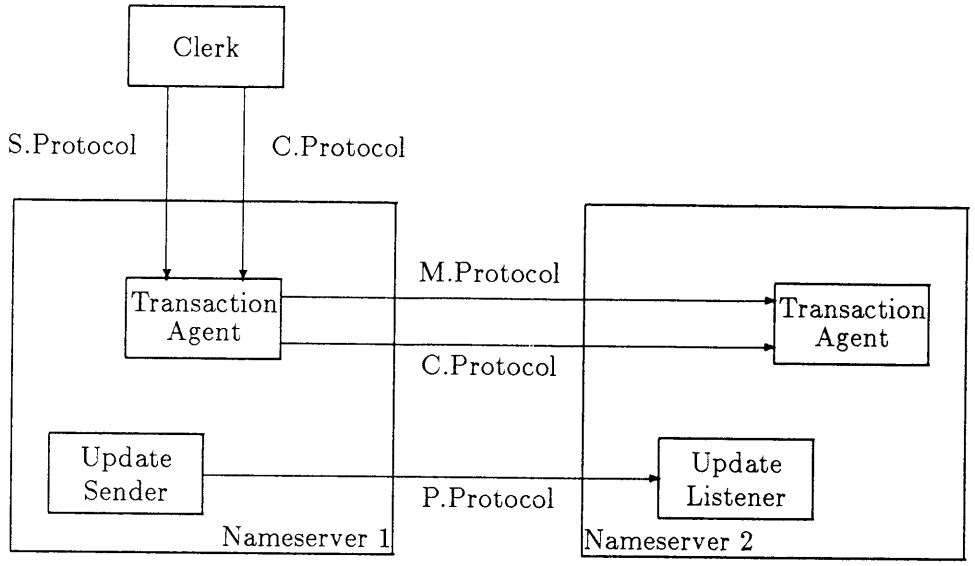
The primary algorithm for producing convergence among the replicas of a directory is called the *Skulker*. Skulks operate independently on each directory in a namespace. Successful completion of a skulk operation ensures that all replicas of a directory have absorbed all updates applied to any replica prior to the time the skulk started. The more frequently skulks are run, the more up to date all replicas of a directory are maintained. Skulks are expensive, however, so there is a tradeoff between resources (bandwidth, processing, memory) and timeliness. Clients can control the frequency of skulks either by adjusting an attribute of a directory, or initiating skulks themselves.

A typical skulk operates as described below. The master replica and all secondary replicas of a directory are linked together into a *virtual ring*, using the *RingPointer* attribute of each replica. The ring keeps multiple skulks of a single directory from getting in each other's way. Each skulk of a directory does the following:

- **Gathers** up all updates made (to the master replica and any secondary replicas) since the last skulk and applies them to the clearinghouse where the skulk is running,
- **Spreads** all the gathered updates to all replicas of the directory, and
- **Informs** all replicas of the timestamp of the latest update all of them are guaranteed to have seen.

8.5 Functional Decomposition of the Naming Service

The Naming Service is functionally divided into a number of major modules. Figure 8.2 illustrates the relationships among the modules of the Naming Service and the protocols employed for communication among the modules.



Legend:

S.Protocol	=	Solicitation and Advertisement Protocol
C.Protocol	=	Clerk–server Protocol
M.Protocol	=	Directory Maintenance Protocol
P.Protocol	=	Update Propagation Protocol

Figure 8.2: Naming Service Modules and Protocols

A Naming Service client accesses the Naming Service through a *client interface.* A summary of the functions available to clients of the Naming Service is provided in Table 8.3.

8.5.1 Clerks

The client interface is provided by a Naming Service Module called a *clerk.* The clerk is the only module of the Naming Service that must reside in all systems; other modules reside only in nameservers. The clerk ascertains the appropriate nameserver to process a request, invokes the *Clerk-Server Protocol (C.Protocol)* to speak to as many Nameservers as necessary to satisfy the request, and contains a cache of recently accessed entries to improve performance. The relationship of the clerk to a nameserver is illustrated in Figure 8.2.

Clerks are responsible for learning about at least one nameserver that can either pro-

Function	Description
EnumerateAttributes	Enumerates the attributes of an object entry, directory entry, soft link, or clearinghouse
ReadAttribute	Returns the value(s) of the specified attribute
ModifyAttribute	Modifies (or deletes) an attribute or attribute value
TestAttribute	Tests an attribute for whether a specified value is one of its current values
CreateObject	Adds an object entry to the namespace
EnumerateObjects	Returns the names of object entries in a directory
DeleteObject	Removes an object entry from the namespace
CreateDirectory	Creates a child directory of the specified parent directory in the namespace
AddReplica	Adds a Clearinghouse to the replication list of a directory
RemoveReplica	Removes a clearinghouse from the replica set of a directory
DeleteDirectory	Removes the specified directory from the namespace
EnumerateChildren	Returns information about child directories of the specified parent directory
Skulk	Skulks a directory to force convergence of its replicas
CreateLink	Creates a soft link for an entry
DeleteLink	Deletes a soft link for an entry
EnumerateLinks	Enumerates the Soft Link entries in a directory
ResolveName	Follows a chain of soft links and returns the full name of the entry pointed to. Cycles are detected.
TestGroup	Tests for group membership, allowing for recursively defined groups and detecting cycles

Table 8.3: Summary of Naming Service Client Interface

cess a request or provide information about other nameservers that may be useful. This initial information is obtained via the *Solicitation and Advertisement Protocol (S.Protocol)*. Nameservers periodically advertise their availability.

8.5.2 Nameservers

Nameservers consist of four modules. The *Nameserver Control module* provides the Management Interface and coordinates the overall operation of a nameserver – turning it on and off, bringing clearinghouses online and offline, etc. This module is also responsible for advertising the availability of the nameserver to any clerks that may be listening.

The *Transaction Agent module* processes the nameserver side of the Clerk-Server Protocol, performing operations requested by clerks. It accesses clearinghouses, performing creations, updates, deletions, and lookups as specified by the clerk. It causes modifications for an entry to be propagated to all replicas of the directory when a change is made.[3] The Transaction Agent is also responsible for coordinating the creation, deletion, and modification of directories using the *Directory Maintenance Protocol (M.Protocol)*, communicating with other Transaction Agents as necessary.

The *Update Sender module* is responsible for propagating any changes made by a Transaction Agent to all clearinghouses that need to know about the change. It uses the *Update-Propagation Protocol (P.Protocol)* to accomplish this. The *Update Listener Module* receives updates sent by the Update Sender and records the changes in the appropriate clearinghouse. The Update Sender and Update Listener are also responsible for ensuring that clearinghouses can find each other when a new clearinghouse is created or if a clearinghouse moves. This is accomplished by clearinghouses modifying their own registration information in the namespace and by maintaining appropriate pointers such that the root of the namespace may be found by starting at any clearinghouse in the namespace.

[3]The actual propagation is carried out by the Update Sender module.

Chapter 9

Support of X.25

Public data networks offering the X.25 interface are available in many countries and represent an economical alternative to leased and dial-up lines for many applications. CCITT Recommendation X.25 defines a standard interface between a system, termed the *Data Terminal Equipment (DTE)* and another system, termed the *Data Communication Equipment (DCE)*.

The DCE is an access point to a *Packet Switched Data Network (PSDN)* implementing the X.25 interface. The X.25 interface is structured into three levels:

- *Level 1*, the *Physical* level, defines the characteristics of the physical link between the DTE and DCE and resides in the Physical layer of DNA. The X.25 Recommendation defines Level 1 only by reference to the X.21 standard for digital networks and an interim X.21*bis* standard for access via analogue lines; X.21 is discussed in Chapter 2.

- *Level 2*, the *Frame* level, defines the mechanisms for reliable communication between the DTE and the DCE. This level resides in the Data Link layer of DNA. The LAPB protocol[1] is used at Level 2.

- *Level 3*, the *Packet* level, defines the format and meaning of the data field contained in each frame. The remainder of this Chapter describes Level 3 facilities and the functions they provide in DNA.

9.1 X.25 Service Functions

Levels 1 and 2 must be operational before packets can be sent on an X.25 virtual circuit (Virtual Circuits are defined in 9.3.1.2) between a DTE and DCE. Level 3 functions are defined in terms of a set of procedures, formats and optional facilities supported at the packet level user interface of a DTE (operating in conformance with the X.25 Recommendation). These include functions for the establishment and release of circuits, the management of circuit characteristics and packet flow-control. The Level 3 functions that are specified in Phase V reflect the utilization of the X.25 packet level protocol in Phase V:

[1]LAPB is a subset of the *High-level Data Link Control (HDLC)* protocol.

- To provide X.25 packet level communication facilities to user-written programs that reside in the DNA Application layer. These facilities allow a user program in a DNA system to communicate with both DNA and non-DNA systems over an X.25 PSDN.

- To provide the Transport layer with a set of capabilities to support the *Connection-Mode Network Service (CONS)* over an X.25 PSDN by mapping between the abstract primitives and parameters of CONS, and X.25 procedures and packet types. The CONS abstract primitives include those for network connection establishment, network connection release, data transfer and reset.

- To provide the Network layer with basic, underlying capabilities to support the ISO 8473 *Connectionless-Mode Network Service (CLNS)* over an X.25 *subnetwork*. The Network layer manipulates X.25 service functions such that X.25 virtual circuits[2] can be used to realize routing over the subnetwork.

9.2 X.25 Modules

The following modules cooperate to provide the X.25 functions in Phase V DNA.

X.25 Gateway Access Module. This presents an interface to user-written programs on a host DNA system and communicates with a Gateway system (a DNA system that is a DTE on an X.25 PSDN). It resides in the Application layer of the host system.

X.25 Server Module. This resides in the DNA Application layer and presents local user-written programs in the system with an X.25 access interface. When the system is also an X.25 gateway it communicates with an X.25 Gateway Access Module over a DNA Session Control connection.

X.25 Packet Level Module. This provides the functions of the X.25 Level 3 packet level protocol. It is used by the X.25 Server Module to gain access to an X.25 PSDN.

LAPB Module. This provides the functions of the X.25 Level 2 frame level protocol. It resides in the Data Link layer of the system and is used by the X.25 Packet Level Module to achieve reliable, frame level communication with the DCE.

9.3 X.25 Packet Level

Functional descriptions of Level 1 and Level 2 of X.25 can be found in chapters on the DNA Physical layer (Chapter 2) and Data Link layer (Chapter 3) respectively. This section describes the features of Level 3, the packet level of X.25.

The use of X.25 circuits for ISO CONS and CLNS/Routing involves mapping from the abstract requirements of the respective "service" to X.25 circuit characteristics for both outgoing and incoming calls. Some or all of these abstract requirements can be codified

[2]Two DNA systems may communicate through an X.25 PSDN over a *dynamically established data link (DED)* (Section 5.2.4) and use standard DNA protocols and utilities.

into a call parameter profile or *template* (see Section 1.3.1). The user interface to the X.25 packet level in Phase V uses the template as a mechanism that allows the invocation of default characteristics for outgoing calls and discrimination criteria for incoming calls on X.25 circuits. Service specific templates are held in appropriate databases and maintained by network management.

9.3.1 Basic Features of the Packet Level

9.3.1.1 Logical Channels

X.25 circuits are defined in terms of logical channels at Level 3. Each logical channel has a unique identifier in the packet header. The range of channels which can be used by the DTE is specified by the network administrator.

9.3.1.2 Virtual Circuits

The packet level establishes a virtual circuit in response to a user request. Logical channels are used to offer *permanent virtual circuit (PVC)* and *switched virtual circuit (SVC)* user facilities.

A PVC is a permanent association between two DTEs. A DTE can send and receive packets to and from the associated DTE at any time. A logical channel is permanently assigned to each permissible PVC. The number of permissible PVCs is agreed with the PSDN administrator.

An SVC is a temporary association between two DTEs. It is initiated by a DTE sending the network a *call request packet* containing the *destination address*. A logical channel is allocated to the SVC at this time. On receipt of the call request packet the PSDN will send the called DTE an *incoming call packet* on a free logical channel. If the call is accepted the virtual circuit enters the *data phase* and *data packets* can be transferred.

9.3.1.3 Flow Control and Packet Sizes

The PSDN uses flow control in order to control the rate at which it accepts packets.

X.25 has independent flow control for each direction of transfer on a logical channel, implemented at both the DTE and the DCE by employing a window mechanism. The window size represents the maximum number of sequentially numbered data packets that may be outstanding at any given time.

The *More Data* bit (M-bit) indicates to the destination DTE the boundaries of the "original" data packet. A larger packet may be sent from the DTE (or the DCE) as a sequence of consecutive, maximum length packets with the More Data bit set to *on* followed by the last packet, with the More Data bit *off*. As the maximum data packet size can be specified independently at each end of a circuit, their difference can cause the number of packets arriving at the destination to be greater or less than the number originally sent.

The window and packet sizes can remain fixed as agreed at subscription time or an optional facility called the *Flow Control Parameter Negotiation* allows negotiation of window and packet sizes on a per virtual circuit basis.

9.3.1.4 Interrupting

The normal packet sequence can be bypassed by use of *interrupt packets*. Interrupt packets are non-sequenced, carry up to 32 octets of user data and can be delivered even when the destination DTE is not accepting normal data packets.

The initiating DTE receives an *interrupt confirmation packet* for every interrupt packet it sends, implying that the interrupt has been received and acknowledged at the destination DTE/DCE interface. Each interrupt packet must be confirmed before another is sent.

9.3.1.5 Resetting

The reset facility is used to reinitialize a PVC or an SVC in the data phase. It removes all data and interrupt packets, (for both directions of that circuit), that may be in the PSDN. Reset can be initiated by either DTE or by the PSDN by sending a *reset request packet*. The initiating DTE or DCE receives a *reset confirmation packet* to indicate resetting has completed.

Resetting can cause the loss of packets that have been previously acknowledged at the local interface. Therefore the user of Level 3 has to participate in the recovery and re-synchronization following a reset.

9.3.1.6 Clearing

A called DTE has the choice of accepting or rejecting the call; the latter by using a *clear request packet* . (The calling DTE will then receive a response indicating whether or not the called DTE has accepted the call.)

Either DTE may clear an established call by sending a clear request packet. In response the DCE sends a *clear indication packet* to the remote DTE. The remote DTE replies with a *DTE clear confirmation packet* to the DCE which then sends a *DCE clear confirmation packet* to the DTE that initiated the clear. The logical channel at either end is deassigned after the sending or receipt of the respective clear confirmation packet.

9.3.1.7 Restarting

A restart is used to clear all virtual circuits at the DTE/DCE interface. A *restart indication packet* from the DCE forces all virtual circuits at a DTE to be cleared. A DTE sends a restart indication packet to the DCE as part of its initialization procedure.

9.3.2 Optional Facilities of the Packet Level

The X.25 Recommendation defines a number of *Optional User Facilities* for use between a DTE and DCE, enhancing the features available to user programs. Some of these facilities are briefly described below.

The X.25 Recommendation categorizes some optional user facilities as being applicable on a per virtual call basis. Others have to be agreed with the PSDN administrator and therefore remain in force for the duration of the agreement with network management.

9.3.2.1 Closed User Groups

The CUG facility enables the formation of groups of DTEs. Access to a DTE is limited to members of the groups to which the DTE belongs. A CUG allows various combinations of accessibility of member DTEs, including; outgoing access allowed, incoming access allowed, outgoing calls barred, incoming calls barred and selection of a CUG on a per call basis.

A special case called the *bilateral CUG* enables pairs of DTEs to access each other while limiting access to or from other DTEs that have not made bilateral associations.

9.3.2.2 Call Redirection

This facility allows redirection of calls destined to a DTE that is out of order, busy or that has an explicit request set up to redirect all calls. Additional capabilities includes specifying a list of alternative DTEs to try, and specifying a logical chain of DTEs for continued redirection.

9.3.2.3 Network User Identification

This facility allows the DTE to provide information to the PSDN for security, network management or billing purposes on a per call basis. The DCE can clear a call on the basis of unsatisfactory identification.

9.3.2.4 Call Charging

A DTE which subscribes to *Reverse Charging Acceptance* agrees to accept incoming calls which request reverse charging. A DTE subscribing to *Local Charging Prevention* optional facility prevents the establishment of calls which the user must pay for. *Charging Information* is a facility that allows the DCE to indicate to the "DTE to be charged", information that will allow it to calculate the charge. It can be applied on a per call basis or can be subscribed to apply to all calls for which the DTE will be charged.

9.4 X.25 Gateway Access

The *X.25 Gateway Access* facility enables programs in any host DNA system, containing a Gateway Access Module, to gain access to an X.25 PSDN. It also enables incoming calls to be directed to a user process residing at the host DNA system.

Gateway operation is accomplished by establishing a connection between the Gateway Access Module and a DNA system that is a DTE on an X.25 PSDN. Such a DNA system is termed the *Gateway system.* Programs using an X.25 gateway can access all the facilities available at the DTE. Figure 9.1 illustrates the X.25 Gateway Access facility.

The communication between the Gateway Access Module and the Gateway system is over a DNA Session Control connection and is based on the *Gateway Access Protocol (GAP)* .

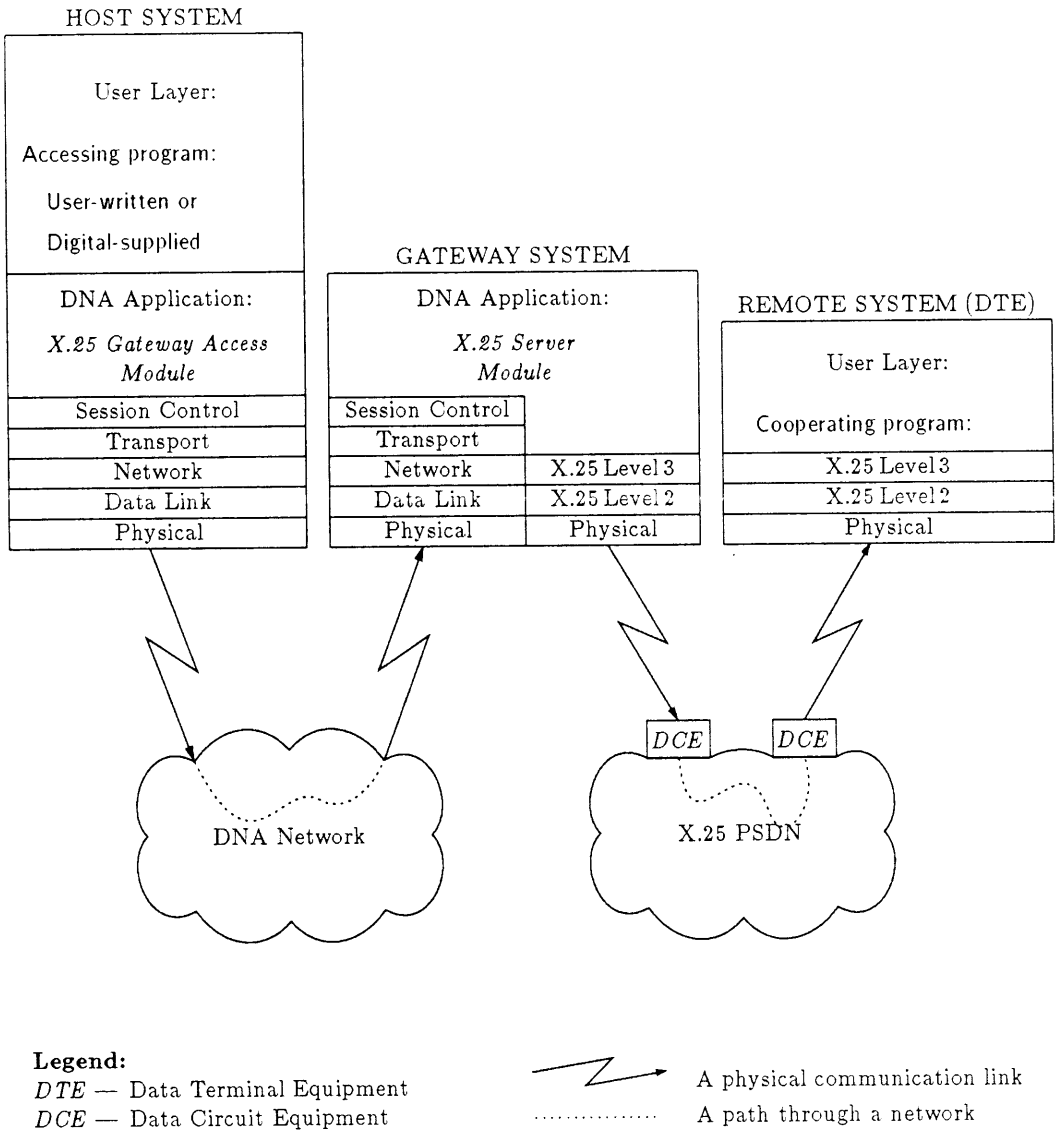


Figure 9.1: The X.25 Gateway Access Facility

The Gateway Access Protocol defines messages that accomplish remote access of the X.25 Level 3 service interface functions. Messages are sent and received on the Session Control connection between the host system and the Gateway system.

The GAP message-types reflect functions available at the Level 3 interface and include those for:

- Opening a PVC
- Making outgoing calls
- Accepting and rejecting incoming calls
- Clearing calls
- Resetting calls
- Transferring data
- Sending and receiving interrupt data
- Indicating failure conditions on a virtual circuit

9.4.1 X.25 Gateway Access Operation

A GAP dialogue between an X.25 Gateway Access Module and a Gateway system over a Session Control connection begins with a user-written program opening a *port* (by placing an Open Port call to the Gateway Access Module). It ends with the closing of the port by the program.

Service calls by a user-written program to the X.25 Gateway Access Module are translated into GAP messages and sent to the Gateway system. Similarly, events at the packet level interface of the Gateway system are translated into GAP messages to be received by the Gateway Access Module. Three processes are involved in the X.25 Gateway operation:

- The User process, in a host DNA system, that accesses X.25 Level 3 functions by issuing calls to the X.25 Gateway Access Module (resident in the same system).
- The X.25 Gateway Server process, in the Gateway system, that exchanges GAP messages with the X.25 Gateway Access Module (over the Session Control connection with the "host" system).
- The Remote DTE process on the remote X.25 system with which the Gateway system communicates over an X.25 virtual circuit.

When a port is opened by the User process, information in an associated template specifies whether the User process requires a PVC, wishes to make an outgoing call or wishes to receive incoming calls. Opening a port causes a Session Control connection to be established and the appropriate GAP message to be sent to the Gateway system.

Subsequent GAP messages depend on what was invoked. For example, a successful outgoing call results in a message that indicates confirmation of the remote system having

accepted the call. Data transfer can then proceed between the User process and the Remote DTE process on the remote system (as can other operations that are valid while in the X.25 data phase).

Either the User process or the Remote DTE process can initiate shut down of the association. Failures at the PSDN can also result in loss of connection between the two processes. The Gateway Access Module reports such events to the User process.

Chapter 10

Network Management

DNA network management allows system or network managers to control and monitor the operation of a DECnet network. It allows network operating parameters to be configured where configuration is necessary, or for fine tuning of these parameters following auto-configuration. It allows the manager to startup and shutdown network components as needed. It allows the network to be monitored, providing information related to network traffic and performance. This data, collected in real time, can be reduced offline to produce statistical and auditing information. It may also play a role in long term network planning. DNA network management can provide information warning network managers of faulty or failing network components, both hardware and software. The facilities provided allow network problems to be detected. Following detection of a problem, a faulty component can be isolated from the network, returning to service once repaired.

DNA network management is developed within the framework of a distributed system management *model.* This design approach provides the following benefits:

- Network management is consistent across the architecture, which in turn allows products developed by different sources to be managed in a consistent way.
- Network management is modular, which allows systems to be as simple or as complex as appropriate to the services they provide their users.
- Network management is extensible, which allows new functions to be added to DNA and managed consistently with existing functions.

DNA fully defines the network management capabilities of each architectural layer or module.

Management of a DECnet network can either be distributed or centralized. Where it is distributed, the extent to which it is distributed can be controlled. A DECnet network can be managed by a human operator at a terminal, or by specialized network management programs. DNA defines a set of primitive management functions which can then be combined and tailored to meet the specific requirements of a particular network configuration or administration.

DNA network management is designed to utilize other DNA services using standard interfaces, for example the DNA Naming Service or the Session Control layer. In general,

management functions are performed in the upper layers of DNA. The services of the Data Link layer are used directly to allow basic management operations such as loading and dumping when not all network protocols are operating, for example during system initialization or failure.

DNA Phase V is based upon the emerging international standards for network management. However, the current draft proposals do not cover all aspects of managing networks. In cases where standards are not currently being developed, DNA uses proprietary solutions with the intention to migrate towards international standards when these become available.

10.1 DNA Network Management Model

DNA network management is designed within the framework of a distributed system management model. The model defines in a general way the management components in a distributed system and the interfaces between them. There are two major components: *directors* and *entities*. Directors are defined as the management software used by a network manager. Entities are the components that make up the network and which are managed. Figure 10.1 illustrates the model.

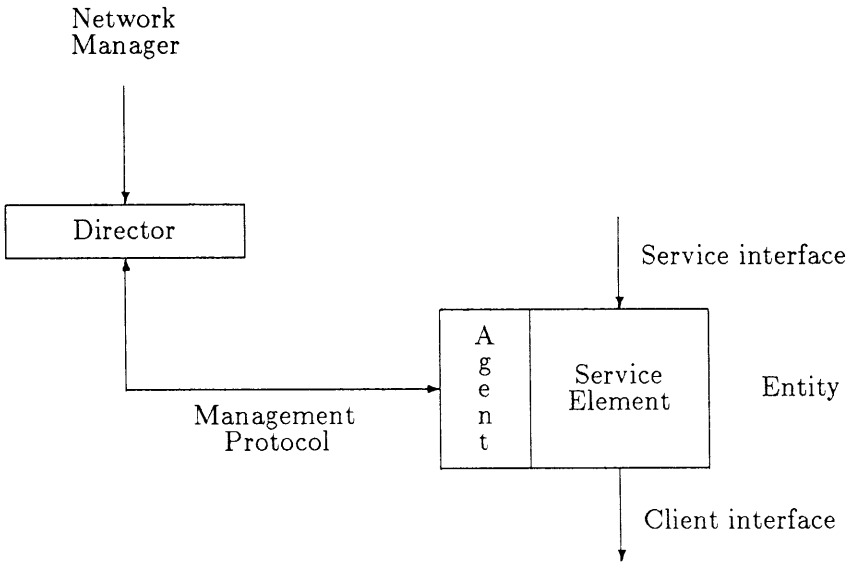


Figure 10.1: DNA Network Management Model – Functional Components

DNA network management is used through a director. Directors, if they are to be used by human operators, provide a user interface for this purpose. They interpret commands entered at a management console, send these commands to a target entity for processing and present the results back to the console. An entity has two logical parts: a *service element* and an *agent*. The service element performs the primary functions of the entity, for

example, provision of a layer protocol, while the agent provides the management interface to the entity.

All directors use common mechanisms to communicate with the entities they manage. The same director can manage local entities, and remote entities on many different systems. If the managed entity resides on the local system, the director may use a local interface to access it. If it resides on a remote system, a *management protocol* is used.

The management information and operations that pass between directors and entities are described below:

Directives. Directives are the management commands issued by a director to an entity. They allow a director to read and alter an entity's management attributes, or to request it to perform a specific action.

Attributes. An attribute is a piece of management information maintained by an entity. Each attribute has a name allowing it to be accessed by management. There are a number of different attribute types, listed below:

- *Identification.* An identification attribute identifies an entity to network management.
- *Characteristics.* Characteristic attributes allow a manager to control the operating parameters of an entity. For example, the parameters which determine the DDCMP polling rate, or the cost of a Routing circuit. In general, characteristics take default values when the entity is created, but these values may subsequently be changed only by network management commands.
- *Status.* Status attributes allow a manager to inspect the current state of an entity. Unlike characteristic attributes, status attributes may change without management intervention, for example, in a protocol entity to reflect the changing state of a connection.
- *Counters.* Counters are attributes whose value indicates the number of times an operation has been performed by an entity, or a particular condition has been detected.

Events. An event is an occurrence of a specific normal or abnormal condition. For example, the discarding of a data NPDU by the Network layer because the destination is unreachable.

10.1.1 Entity Hierarchy and Naming

A DECnet network is constructed out of many manageable components – the more systems in the network, the more manageable components there are. To allow management of large networks, the components must be organized into a logical structure, and named in a way that allows a manager to deal with the complexity. The organization defined in DNA is a hierarchical one with parent entities "above" their children in the hierarchy.

Each computer system in a DECnet network is a top-level component which is represented to network management by the *node* entity. A node entity is assigned a name which

is unique throughout the network. This forms the basis for assigning names to "lesser" entities in the system which will themselves have network-wide uniqueness. Node names are registered with the DNA Naming Service together with an address by which the system can be reached.

Below the node entity are *module* entities. Modules comprise a group of network functions which together provide a particular service. For example, DNA defines Routing, OSI Transport and DDCMP modules, and many others. There is only one occurrence, or *instance* of a module entity in a node. For this reason, modules can be uniquely identified within a node by their *class name,* for example "DDCMP", and no further qualification by *instance name* is required.

At the next level down in the hierarchy are subordinate entities, or *sub-entities.* These are defined to allow management of some part of a module's functions. The DDCMP module, for example, maintains Link sub-entities for each communications link over which the protocol operates. "Link" is the class name of the sub-entity. Each instance of the Link sub-entity requires further identification, to allow it to be distinguished from the others. Further levels of sub-entity can be defined for each entity.

The entity hierarchy is used to form an *entity name.* An entity name comprises a *global* part and a *local* part. The global part consists of the node entity name which identifies the node in the network. The local part identifies the entity within the node. The local name is a derived from the class name and instance name (if applicable) of the entity itself and those of its parent entities, up to the level of the node entity. This is similar to a fully specified mailing address, where the hierarchy might consist of the country, state, city, street, house and individual.

A directive for an entity in a remote system is sent to the node agent of the system and includes the entity name. The node agent uses the local part of the name to identify a sub-entity to handle the directive. This process may be repeated by sub-entities on any remaining portion of the name until the target agent is reached.

The entity hierarchy is defined to allow network components be uniquely identified for management purposes. It imposes no restrictions on the actual management organization of a DECnet network.

10.2 Network Management Operation

This section gives an overview of the major Phase V network management components and operation.

10.2.1 Common Management Information Protocol (CMIP)

The DNA *Common Management Information Protocol,* or *CMIP,* provides an encoding of the network management operations which can be performed on entities, and their parameters. DNA CMIP is based closely on the emerging ISO CMIP standard, currently at a draft stage. It is in fact the combination of two protocols – the *Management Information Control and Exchange (MICE) protocol,* used between a director and a node agent, and the *Management Event Notification (MEN) protocol.* DNA CMIP is a simple Application layer

request-response protocol which operates over DNA Session Control. It replaces the Network Information and Control Exchange (NICE) protocol of DNA Phase IV, and provides operations to get and set management attributes, to request execution of management actions and to report events.

10.2.2 Event Logging

DNA provides mechanisms which enable the event information generated by entities to be distributed to points in the network where it can be stored or analyzed. DNA Phase V event logging is an extension of the facilities available in Phase IV, and consists of the following components:

- Sources
- Sinks
- Event dispatchers (EVD)

Event sources are the DNA entities that detect events and initiate the production of event reports. *Event sinks* are the consumers of event information. They are entities that process this information, or simply store or display it. Phase V event logging allows an arbitrary number of sinks, possibly with each providing different features.

An *event dispatcher*, or *EVD,* serves as an intermediary between sources and sinks. Each node in the network, if it reports events, has an EVD. Entities in the node post event records to the local EVD which is responsible for buffering the information and distributing it to event sinks. Network management can control the operation of an event dispatcher. *Event streams* may be created which define the sinks for event information, and *event filters* which define the event subset reported to each sink. Further filtering of events may be carried out at the sink. This permits certain event sinks to be restricted to the processing of particular events. Event generation may be turned off at the source. The Management Event Notification protocol is used to carry event reports between an EVD and an event sink.

10.2.3 Maintenance Operations Protocol (MOP)

Maintenance operations are special, primitive functions that must be available in a system without the services of the higher layers of DNA, for example if the system is initializing and not fully operational. DNA defines a simple management protocol for use in these circumstances called the *Maintenance Operations Protocol,* or *MOP.* MOP is a client of the Data Link layer, requiring only minimal data link protocol support; the MOP modules handle all message acknowledgment, time-out and retransmission functions. The following maintenance functions are defined:

Down-line loading. The down-line load function allows a memory image to be sent to a directly connected, or adjacent system. If the image is that of a program, the function allows program execution to be started at a specified memory address following the

load. On a CSMA/CD data link, a system can multicast its load request and select the first system that responds as the server.

Up-line dumping. The up-line dump function allows the contents of memory to be sent to an adjacent system. On a CSMA/CD data link, a system can multicast its dump request and select the first system that responds as the server.

Link loopback testing. MOP tests communication links by looping a test message at various points in the physical connection. By moving the loopback point and isolating components, a manager can diagnose link problems.

System console control. MOP can be used to control remote, possibly unattended systems through emulation of a console terminal. This allows the remote system to be restarted using a "Boot" command.

10.2.4 Network Control Language (NCL)

DNA defines a command line interface to directors, called *NCL*. NCL is new to DNA Phase V, replacing the Network Control Program (NCP) of Phase IV. NCL gives access to the directives defined for all of the entities in DNA.

NCL specifies the general syntax rules for network management commands, and for display of subsequent responses. An implementation of NCL takes input from a terminal and causes directives to be issued to entities using MICE protocol messages. The NCL command syntax consists of a verb, entity name and a sequence of argument identifiers or identifier-value pairs. The following examples illustrate the syntax:

```
Set Node Engineering.Reading Routing Circuit DMC-1 L1 Cost 10

Show Node Engineering.Reading DDCMP Link DMC-1 Counters
```

NCL permits the use of *wildcarding* at various points in the command. This can be used to effect a management operation at many instances of an entity class on a single node. For example, if `DMC-1` was replaced by * in the `Show` command above, all DDCMP Link entities on node `Engineering.Reading` would return their Counter values.

NCL provides commands that invoke calls to the DNA Naming Service, for example to register a node in the namespace. Additionally, NCL permits management of a Phase IV system from a Phase V system. Phase IV style NCP commands are introduced by the keyword "`NCP`".

Chapter 11

DNA Applications

DNA supports a wide range of applications, from general purpose network-wide applications such as file transfer and virtual terminals, to highly specific distributed applications written to solve particular customer problems. Digital currently offers a large number of widely-used application protocols with DNA. This chapter briefly describes some of those application protocols. Note that *all* Phase IV applications will continue to run on Phase V systems.

11.1 Heterogeneous File Access and Transfer

DNA provides the *Data Access Protocol (DAP)* for accessing and transferring files in a heterogeneous DECnet network. DAP provides the following functions and features:

- Supports heterogeneous file systems.
- Retrieves a file from an input device (disk file, terminal, etc.).
- Stores a file on an output device (magtape, printer, disk file, etc.).
- Transfers files between systems.
- Supports creation, deletion, and renaming of remote files.
- Lists directories of remote files.
- Recovers from transient errors and reports fatal errors to the user.
- Allows multiple data streams to the same remote file.
- Submits and executes remote command files.
- Permits sequential, random, and indexed (ISAM) access of records.
- Supports wildcard file specification for sequential file retrieval, file deletion, file renaming, and command file execution.

- Permits an optional file checksum to ensure file integrity.

DAP is designed to minimize protocol overhead. For example, the file transfer mode eliminates the need for DAP control messages once a data file flow begins. Also, small file records can be blocked together and sent in one large message.

Two cooperating processes exchange DAP messages: the user process and the server process that acts on the user's behalf at the remote system. User I/O commands accessing a remote file are mapped into equivalent DAP messages and transmitted via a transport connection to the server at the remote system. The server interprets the DAP commands and actually performs the file I/O for the user. The server returns status and data to the user.

11.2 Network Virtual Terminals

The Network Virtual Terminal Service consists of the following:

- A functional model of a terminal (virtual terminal) for access through the network. This model is known as the network command terminal.
- A set of protocols for communicating between a host system, with an application program running, and a server system, with a terminal directly attached.

The Network Virtual Terminal Service provides these functions and features:

- Distributes terminal handling functions between two systems.
- Supports heterogenous host systems. Different operating systems can cooperate via common protocols, and a host operating system can manage a terminal in its own way, regardless of which operating system runs in the server.
- Allows a server to connect to a specified host or a host to a specified remote terminal.
- Provides terminal input/output and characteristics management functions at the operating system services level. These include:
  - Accept input even if the program has not issued a read request (typeahead).
  - Take action on certain characters immediately as keys are struck (*out-of-band* character processing).
  - Recognize ANSI standard escape sequences on input and output.
  - Read and set terminal device characteristics.
- Offers standard terminal services, featuring good performance and device independence. Optionally, offers methods of controlling the terminal's behavior in considerable detail.

The virtual terminal service is realized by a pair of protocols, *CTERM* and *FOUND*. FOUND provides connection management and a transparent data transport capability over which a number of terminal usage models may co-exist (for example: command mode, editor mode, forms mode). CTERM implements the model of a command terminal which is the common mode of access to command language processors such as DCL (Digital Command Language).

11.3 Electronic Mail

11.3.1 Mail-11

The Mail-11 protocol is a DNA application protocol which provides a personal electronic mail capability. It allows people to exchange text messages among VMS™, ULTRIX™, RSX-11M-PLUS™, and RSTS/E™ systems. The Mail-11 protocol is implemented on VAX/VMS™ by the *Mail* utility, which also handles the user interface to mail on the local host. Mail-11 is implemented on RSTS/E and RSX-11™ via the *DECMail-11*™ product.

Gateway products provide interfaces from the Mail-11 protocol to other protocols including:

- VAX PSI Mail, which permits mail to be sent between distinct DNA networks using the VAX PSI X.25 communications product.
- Message Router™ (see Section 11.3.2), which permits users of the VAX/VMS Mail utility to gain access to the enhanced capabilities of Message Router including the ability to send mail using the X.400 suite of protocols.
- Ultrix Mail, which permits mail to be sent not only to Ultrix systems in the same DNA network, but also to other systems utilizing the TCP/IP SMTP Mail protocols and UUCP mail.

11.3.2 Message Router

The Message Router, together with its family of layered products, provides a highly reliable store and forward message handling system for the delivery of interpersonal mail based upon the X.400 series of international standards for message handling systems.

Included in the Message Router product family are mail user agents, which provide message transmission and delivery services to users, message transfer agents, which provide store and forward message transport in a DECnet network, and gateways, which provide access to other mail systems and other networks.

Together these components provide common interpersonal mail services to users of multiple vendors' computer systems interconnected by multiple network products.

11.4 SNA Interconnect Applications

DNA provides a rich set of application protocols for communicating with systems conforming to IBM's *Systems Network Architecture (SNA)*. The following subsections briefly

describe some of these SNA-oriented applications.

11.4.1 SNA Gateway Access

SNA Gateway Access provides the following functions and features:

- Supports communication between user-written and Digital-written programs in a host DECnet system, and modules in an IBM host system, over an SNA network.
- Supports programs in DECnet systems which act as SNA Secondary Logical Units (SLUs) of any of the defined SNA Logical Unit Types (LUs).
- Allows user programs full access to the facilities provided by the SNA Transmission Control and Data Flow Control layers.
- Communicates with the SNA network as a Physical Unit Type 2 (PU2).
- Permits user-written programs to participate in SNA sessions from any DECnet system containing an SNA Gateway Access module. The user program does not have to reside in the DECnet system that is acting as the PU2 directly connected to the SNA network.
- Permits the user-written program to send messages to and receive messages from the SNA System Services Control Point — that is, the user program has access to the SSCP–SLU session.
- Manages the SNA session pacing automatically for the user program.
- Allows an IBM application to initiate a session with a DECnet partner, by automatically activating the DECnet object when a session request (BIND message) is received from the SNA network.
- Provides an access control mechanism to restrict SNA session address use to authorized users.
- Recovers from transient errors and reports fatal errors to the user.

To use the services of SNA Gateway Access, the user program makes calls to the *SNA Gateway Access module*. SNA Gateway Access communicates with a process called the *SNA Gateway Server* which runs in a DECnet system that is also a PU2 node on the SNA network (the "gateway system"). The communication occurs over DECnet Session Control; SNA Gateway Access Protocol messages are exchanged over the connection to make the facilities of SNA sessions available remotely. Figure 11.1 illustrates the operation of SNA Gateway Access.

SNA Gateway Access provides the basis for a range of Digital products, including 3270 terminal emulation, printer emulation, remote job entry, DISOSS data exchange, file transfer and data access, and various user programming interfaces.

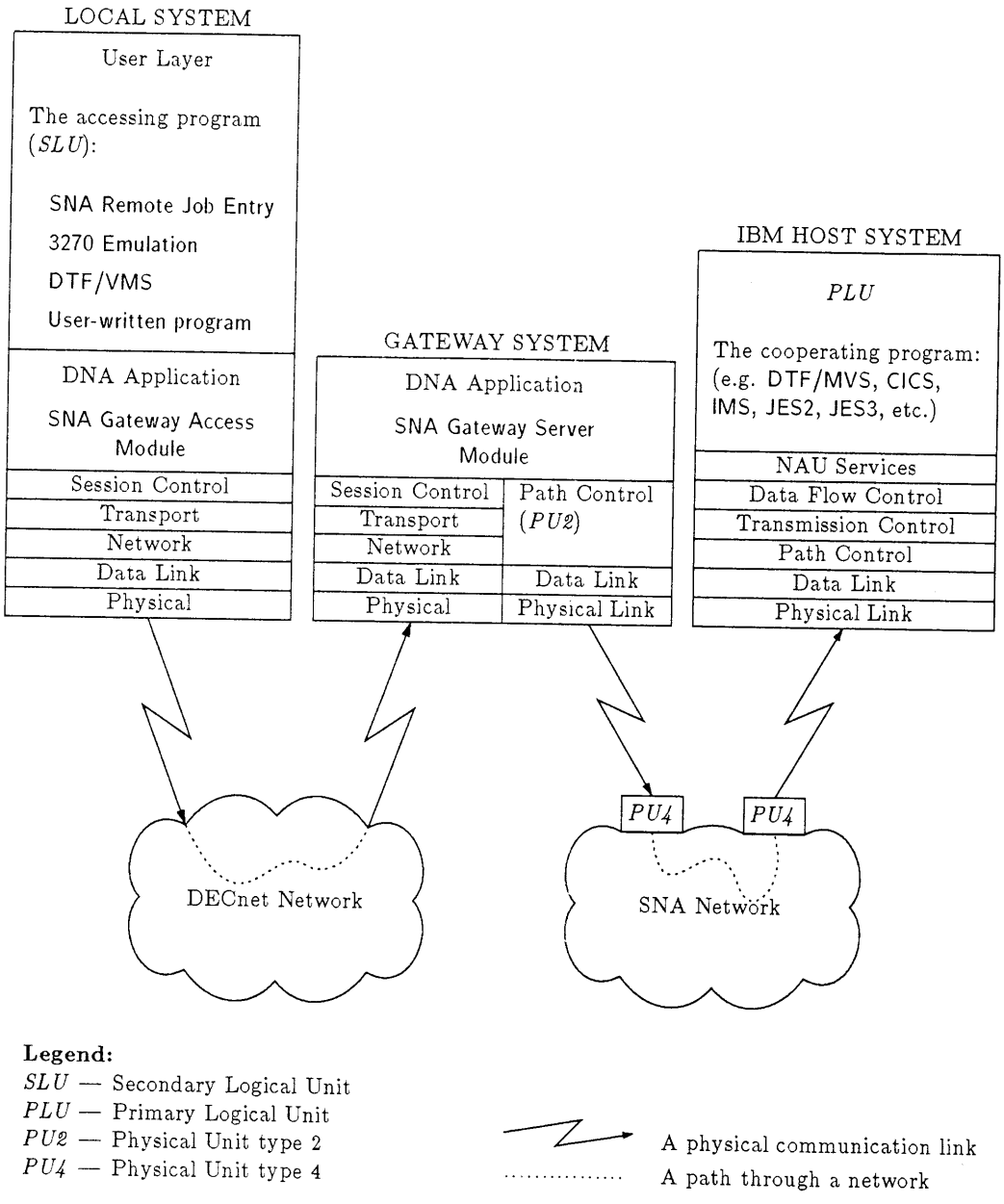


Figure 11.1: SNA Gateway Access Operation

11.4.2 3270 Terminal Emulator

The 3270 Terminal Emulator (TE) mimics an IBM 327x block-mode terminal connected to a 3274 cluster controller, and thereby allows a VAX/VMS user at a VT100™, VT200™, or VT300-class terminal[1] to interactively log on to IBM host applications. The TE handles the conversion to and from the data formats used by 3270-class terminals, manages the efficient updating of the VT terminal screen, and communicates with the IBM host as an SNA Logical Unit Type 2 through the SNA Gateway.

11.4.3 Remote Job Entry

The Remote Job Entry subsystem provides batch access to IBM's MVS operating system. An RJE server, resident in the gateway itself, supports emulation of up to four workstations.

VAX/VMS users can submit JCL files to a queue on a host VAX associated with one of the workstations; the RJE server will eventually transmit the JCL "deck" to the IBM mainframe for processing. Later, print and punch output from the job may be routed back to any system in the DECnet network (the server uses ordinary DECnet remote file access to move its input and output data around).

A VAX/VMS operator utility provides access to the console of the emulated RJE workstation.

11.4.4 Data Transfer Facility

The Data Transfer Facility (DTF) software provides for bidirectional file transfer between VAX/VMS and IBM's MVS system. File transfers can be requested from the VAX and from the IBM machine. In addition, user-written programs in the VAX/VMS system can access IBM files at the record level, using standard RMS calls.

The DTF software includes VMS-resident components (DTF/VMS) and MVS-resident components (DTF/MVS); both parts are written and supplied by Digital. In the VMS system, DTF/VMS is implemented as a data transfer server which can be used by any client in the DECnet network which supports DAP (see Section 11.1). DTF/VMS communicates with DTF/MVS through the SNA Gateway using the DECnet DAP protocol running on top of a private Logical Unit Type 0 protocol. Thus, unlike earlier SNA Gateway layered products which emulated existing IBM devices, DTF implements a DECnet protocol on the IBM system itself.

DTF/VMS also provides the ability to translate the data records during transfer according to field definitions stored in the Common Data Dictionary (VAX CDD™). Thus, records containing mixed data can be accommodated, as well as simple EBCDIC to ASCII translations.

---

[1] Or a compatible terminal emulator, such as a workstation.

11.5 VMS Services for MS-DOS

VMS Services for MS-DOS is a VAX based remote file server for personal computer systems running Microsoft's MS-DOS operating system. The server allows personal computers, running Digital's Personal Computer Systems Architecture (PCSA) software, to create, read and modify MS-DOS files on a VMS system. The user connects to VMS services (established by the VAX system administrator), via the PCSA client system software, and is presented with the equivalent to a MS-DOS file oriented disk device. File sharing and byte range locking are implemented (according to the Microsoft specification) by the file server.

The server implements the Microsoft Networks (MS-NET) Server Message Block (SMB) protocol, version 1.0. The server provides an MS-NET session level interface; all lower layers (Transport, Network, etc.) are provided by DECnet. The software, although intended to operate in a LAN, will also operate in a WAN environment. The file server may be installed on stand-alone VAX systems, or on VAXcluster™ systems.

11.6 Time Service

The notion of time is taken for granted in most of today's centralized computer systems. However, the mechanisms used for providing time in these systems are inadequate when applied to distributed systems. In the distributed environment, new mechanisms are required. A global notion of time can be extremely useful to co-ordinate the operation of a distributed system or distributed applications.

The *Digital Time Service* is an architecture for providing and maintaining correct time in a distributed system. The architecture conforms to the client-server model of distributed systems whereby a collection of time servers supply the time to clients through intermediaries, called clerks, residing on the same machines as the clients. Since the value of the time at any instant is never known exactly, the time can at best be expressed in the form of an interval which is guaranteed to contain the instantaneous value of the time. By obtaining the time from a number of servers, systems are able to compute a correct time interval even in the presence of some faulty servers. This ensures that the time obtained by clients is correct with arbitrarily high probability. The Time Service architecture specifies the operation of clerks and servers including the methods for obtaining the time from a server, computing the best time interval from a set of intervals, maintaining the local clock correctly, and detecting faulty servers. The algorithms are fully distributed and asynchronous.

11.7 Computer Conferencing

VAX Notes Computer Conferencing allows people throughout a DNA network to participate in round-table discussions, bridging distances between the participants and accomodating time differences in working hours in a global network to enhance information sharing throughout an organization.

Any number of VAX Notes Conferences can be established in a network. Conference participants use the VAX Notes client program installed on their home systems. These client programs communicate with VAX Notes server programs using a DNA application protocol which allows access to conferences located throughout the network for reading and writing conference entries, known as notes.

Conference moderators create conferences for the discussion of a major topic or group of related topics. Each conference can be considered to be logically equivalent to a large bulletin board organized into individual topics, which, continuing the bulletin board analogy, are posted along the top of the bulletin board by the participants. Posting a note at the top of the bulletin board is known as writing a new topic. Participants who wish to add to the discussion of existing topics "reply" to that topic; in the bulletin board analogy this is equivalent to posting a note directly below the existing topic and any existing replies.

The moderator of a conference has special capabilities for controlling the discussion, including the ability to restrict conferences to a membership list, to delete or "hide" notes deemed inappropriate to the discussion or which require further clarification, to change the titles of topics or replies to topics to improve the organization of a conference, to create keywords which may be associated with notes in the conference, and to designate additional moderators.

11.8 VAX System Performance Monitor

The VAX System Performance Monitor (VAX SPM™) uses a DNA application protocol to allow a single system in a VAXcluster™ system to communicate with all other members of the cluster. VAX SPM uses this for starting and stopping data collections on the entire cluster from a single system and for doing live video displays on the cluster.

The VAX SPM DNA application works asynchronously. When performing a cluster command, a connection is established to each system in the cluster. The protocol establishes the links, and sends a request for data to all of the other systems. The other systems send the data and when all the data has been sent, an end of data message is sent. The receiving end keeps track of which systems have completed their data transmission and when all systems have completed, the data is processed.

11.9 Videotex

Videotex is a computer based electronic retrieval system which uses a hierarchical arrangement of information, is video-terminal based, and provides a very simple user interface. VAX VTX™ is Digital's product which provides videotex services on VAX/VMS systems. VAX VTX uses a DNA application protocol for communication between participating systems in a DNA network. VAX VTX organizes information into *pages*, allowing pages to be retrieved from *infobases*, which can be resident on any VAX/VMS system.

VTX users, known as *subscribers*, can display, print, and save VTX pages using simple keypad techniques. Subscribers use a client program, known as a *pad* (packet assembler/disassembler), which runs on their system and communicates with a server process

on the system where the infobase resides. The server provides access to the infobase files, making the infobase available for perusal either by general users, or by specified groups as determined by parameters specified on the server system.

The page information itself can be stored in a variety of display protocols, such as ReGIS™, ASCII, NAPLPS, and PRESTEL, giving VTX a considerable amount of flexibility. VAX VALU™, the VTX sister product, provides limited data input and collection via the use of *form* pages.

11.10 Distributed Queueing

The Vax Distributed Queuing Service (DQS) extends the VMS queue system and increases the availablity of printing services in a distributed environment. DQS allows files to be printed on remote VMS systems which have printing devices attached. DQS also supports the examination and manipulation of print jobs.

DQS can be configured to operate as both a client and a server or as a client only. Any system where a print request originates is called a client. Any system with the attached printer that performs the printing is called a server. Client or server systems may be added in the network at any time without disrupting current operations.

11.11 Remote System Management

The Remote System Manager is used within a DECnet network to aid in the management of VMS and ULTRIX systems. Services provided by RSM include:

- File system backup and restore
- Software distribution and installation

RSM operates using a client/server model. The server is a VMS system which contains the backup schedule and software libraries. A manager, located on the server, schedules backups and software distribution operations for one or more client systems.

RSM uses a DNA application protocol to send commands to an agent located on each VMS or ULTRIX system being managed. Upon receiving instructions from the server, the client's agent may perform a backup using the local backup utility, or install an application.

RSM also uses the DNA MOP protocol (see Section 10.2.3), combined with special RSM applications, to load either the VMS or ULTRIX operating system onto a client system over a local area network.

RSM stores information about RSM clients and servers using the Naming Service, which provides RSM with a network-wide namespace (see Chapter 8).

Chapter 12

OSI Upper Layers and Applications

As the DNA Transport layer and below contain protocols which are standardized as part of OSI, OSI applications can be built which interwork with other open systems.

In DNA Phase V, OSI upper layer components coexist with existing DNA higher layer components as a distinct set of layers and share access to the Transport layer with the DNA Session Control layer (see Figure 1.3). OSI upper layers provide communication for OSI applications. OSI upper layers are not integrated into DNA Phase V as the standards describing them are not yet complete.

The components of the upper layers of OSI fall into two categories: those which are general building blocks from which complex OSI applications can be built, and those which are less general or provide the communications facilities to fulfill specific information processing requirements. Section 12.1 discusses the former components and Section 12.2 discusses the latter.

12.1 General OSI Upper Layer Components

Together, the upper layers of OSI provide the communications services for OSI applications partaking in distributed information processing activities. The services of a given layer are made visible by pass through services to the next higher layer, and so on up through the layers. Each layer also contributes services pertaining to its layer functions. The totality of services passed through from the OSI Session and Presentation layers, together with specific services provided in the Application layer are provided to OSI applications. The OSI upper layers contribute the following general purpose functions:

Session layer. The facilities of the OSI Session layer build on the reliable end to end communication facilities provided by the DNA Transport layer. They provide a means by which communicating OSI applications can control access to certain services, allow synchronization and resynchronization to specific points in the communication, and allow the division of the communication into a number of logical pieces of work.

DNA applications, which require facilities similar to those provided by the OSI Session layer, meet their requirements in an application specific way.

Presentation layer. The facilities of the OSI Presentation layer build on and pass through the services provided by the OSI Session layer and provide, in addition, a means by which OSI applications can agree on representations for data while in transit between communicating partners. Subsequently, the Presentation layer causes the data transferred to be in the agreed representations. This provides OSI applications with a means to transfer information without loss of the application semantics.

These facilities are required in order to support the interworking of applications in heterogeneous networks where the local representation of data may differ between each open system. DNA applications do not require these facilities as they only operate in homogeneous networks.

Application layer. The facilities of the Application layer are subdivided into a number of *application service elements* (ASE), each of which provided a specific set of functions in a reusable way. Some ASEs form the components of specific applications and are described in Section 12.2. General purpose ASEs are described below:

Association Control Service Element. Association control services are provided by the Association Control Service Element (ACSE). These allow OSI users to establish an *application association* between them for the purpose of exchanging information. When information exchange is complete, ACSE provides services to destroy these associations.

12.2 OSI Applications

Several OSI applications are currently under development as part of the OSI set of standards. These standards define the information processing tasks required and the application service elements necessary to fulfill the communications related requirements.

The following specific applications are of interest as they provide similar functions to those provided by DNA applications:

File Transfer, Access and Management Services. FTAM is a four part International Standard which provides the following services:

- transfer of an entire file;
- read, write or modify of existing components (that is, records) of a file;
- file management – creation of files, deletion of files and reading and modification of file attributes.

When information is transferred between open systems, FTAM preserves the semantics and structure of the file data being exchanged.

Virtual Terminal Services. VT consists of a service definition and a protocol specification which provides services that are similar to those provided by the DNA Network Virtual Terminal (NVT) service but in a heterogeneous environment. Specifically, it supports the interactive transfer and manipulation of structured data in a manner which models character cell terminals.

Glossary

access control The management of rights to use resources.

address prefix Any leading portion of an NSAP address.

address resolution The Session Control function which maps from a Naming Service object name to the identifiers of protocols and corresponding addresses which are mutually supported by the local system and the remote system(s) on which the named object resides.

address selection The Session Control function which provides transparent selection of protocols and addresses for Transport Connection establishment based upon the destination object name.

addressing authority The authority responsible for the unique assignment of Network layer addresses within an addressing domain.

addressing domain A level in the hierarchy of Network layer addresses. Every NSAP address is part of an addressing domain that is administered directly by one and only one addressing authority. If that addressing domain is part of a hierarchically higher addressing domain (which must wholly contain it), the authority for the lower domain is authorized by the authority for the higher domain to assign NSAP addresses from the lower domain.

administrative domain A collection of End Systems, Intermediate Systems and Subnetworks operated by a single organization or administrative authority. It may be subdivided into a number of routing domains.

AFI *See* **Authority and Format Indicator.**

agent That part of an entity which provides the interface to network management.

area The group of systems which constitute a single Level 1 routing subdomain.

area address The concatenation of the IDP and LOC-AREA fields of an NSAP address. A system may have more than one area address, but the systems making up an area must have at least one area address in common with each of their neighbors.

asynchronous link A communications link using asynchronous transmission, in which each transmitted character is framed with start and stop bits. The time interval between characters may be of unequal length.

Authority and Format Indicator The part of an NSAP address which indicates the addressing authority responsible for the assignment of the IDP and its format. It also indicates the format (binary or decimal) of the DSP.

balanced mode An HDLC operational mode used in DNA over full duplex links.

bridge A Data Link relay for interconnecting LANs, used to increase the maximum number of stations, maximum distance, and total available bandwidth.

broadcast subnetwork A multiaccess subnetwork that supports the capability of addressing a group of attached systems with a single message.

call reference A unique value used locally to identify a Modem Connect or X.21 call.

call sharing A form of switched line sharing in which many clients have access to the same call on that line.

carrier sense On CSMA/CD LANs, each station waits before transmitting for the *carrier* indication from the Physical layer to go off, indicating that no other station is in the process of transmitting.

CCITT The International Telegraph and Telephone Consultative Committee, the technical committee of the International Telecommunications Union (ITU), responsible for the development of recommendations regarding telecommunications, including data communications.

centralized management A form of network management where management is performed from a single point in the network.

checksum A value transmitted with a message which is computed from the content of the message itself. This value is recomputed by the receiver of the message to detect whether the message was corrupted during transmission.

child directory entry An entry in the Naming Service which points to a child directory of some directory in the namespace.

circuit switching A telecommunications technique involving the dynamic establishment of a physical connection prior to information exchange, and release of the connection following the exchange.

clearinghouse A collection of directory replicas stored together in one location.

client The user of the service provided by a module or layer in the architecture.

CLNS *See* **Connectionless-mode Network Service.**

CMIP *See* **Common Management Information Protocol.**

collision An attempt by two stations on a CSMA/CD LAN to transmit at the same time. The stations will detect the collision and retry the transmission.

Common Management Information Protocol A management protocol which encompasses the MICE and MEN protocols.

communications link The physical medium connecting two systems.

concatenation The process of assembling multiple nPDUs into a single $(n-1)$SDU for transmission.

congestion Congestion occurs when a network, or part of a network, is overloaded and has insufficient communication resources for the volume of traffic.

congestion avoidance A mechanism used to adjust the load on the network to prevent congestion.

connection control The Session Control function concerned with the system-dependent functions related to creating, maintaining, and destroying Transport Connections.

Connection-mode Network Service A network service which operates according to a connection oriented model. Before data can be exchanged, a connection must first be established.

Connectionless-mode Network Service A network service which operates according to a datagram model. Each message is routed and delivered to its destination independently of any other. The Network Layer of DNA provides this type of service.

CONS *See* **Connection-mode Network Service.**

control station In DDCMP multipoint configurations, the station responsible for data link control.

cost A metric used by the routing algorithm. Each link is assigned a cost, and the routing algorithm selects paths with minimum cost.

credit A flow control mechanism whereby the receiver of data tells the transmitter how many messages it is prepared to receive at a given point.

credit window When operating a credit based flow control mechanism, the credit window identifies the range of message numbers which the receiver is prepared to receive at a given point.

CSMA/CD Carrier Sense Multiple Access with Collision Detection, the channel access method used by Ethernet and ISO 8802-3 LANs.

cyclic redundancy check (CRC) An error detection mechanism which involves polynomial division of the data to be transmitted. The coefficients of the remainder are transmitted along with the data, for checking by the receiver.

DA *See* **Dynamic Assignment.**

DAP *See* **data access protocol.**

data access protocol A DNA proprietary application protocol for accessing and transferring files across a DNA network.

data circuit equipment The equipment that provides the functions required to establish, maintain and terminate a connection between DTEs using a physical circuit or a virtual circuit.

data link connection The Data Link layer communications path between two directly connected systems.

Data Link layer The layer of DNA which provides a communications path between two directly-connected systems in a network.

data link protocol The coding rules and communication procedures which allow the data link service to be provided.

data terminal equipment The equipment, typically a computer system or terminal, comprising a data source and sink connected to common carrier communication facilities.

datagram A frame that is processed by the network independent of any other frames. Datagrams may fail to be delivered without notice.

DCE *See* **data circuit equipment.**

DCM *See* **Dynamic Connection Management.**

DED *See* **Dynamically Established Data Link.**

Digital Data Communications Message Protocol (DDCMP) A Data Link layer protocol used in DNA.

directive A management request sent by a director to an entity.

director The management software used by a network manager.

directory A container in the Naming Service which stores a set of names.

distributed management A form of management where network managers and directors are dispersed across many systems.

distributed system management model The framework within which DNA network management is designed.

Domain Specific Part The part of an NSAP address assigned by the addressing authority identified by the IDP.

DPDU Data Link Protocol Data Unit.

DSAP Destination Service Access Point, the one byte field in an LLC frame on a LAN that identifies the receiving Data Link client protocol.

DSP *See* **Domain Specific Part.**

DTE *See* **data terminal equipment.**

Dynamic Assignment The use of a Dynamically Established Data Link by the Network layer, in such a way that a connection is only made when data is required to be transferred, and the subnetwork address to which the connection is made is determined by the destination NSAP address.

Dynamic Connection Management The use of a Dynamically Established Data Link by the Network layer, in such a way that a connection is only made when data is required to be transferred.

Dynamically Established Data Link A connection oriented subnetwork used by the Network layer.

end system A system which transmits and receives NPDUs from other systems, but which does not relay NPDUs between other systems.

entity A network component that can be managed using network management.

entity attribute A piece of management information maintained by an entity. DNA defines Identification, Characteristic, Status and Counter attributes.

entity hierarchy The logical structure of manageable components in a system.

entity instance An occurrence of an entity.

entity name The name of an entity consisting of a global name part and a local name part.

Ethernet A CSMA/CD LAN similar to the one defined by ISO 8802-3.

EVD *See* **event dispatcher.**

event An occurrence of a normal or abnormal condition detected by an entity and of interest to network management.

event dispatcher The intermediary between event sources and event sinks.

event sink An entity which is a consumer of event reports.

event source An entity that detects events and generates event reports.

extended sequence numbering An option in HDLC which permits sequence numbers up to 127 to be used rather than up to seven.

flooding, network layer A means of propagating a message throughout a network. The message is transmitted to each neighboring system except the one from which it was received. This forms the basis of the mechanism by which Link State PDUs are propagated by the Network layer.

flow control A protocol mechanism used during data transfer to match the relative speeds of transmitter and receiver.

frame A Data Link Protocol Data Unit: a block of data supplied by the Data Link user together with the Data Link envelope.

frame level Level 2 of the CCITT X.25 Recommendation which defines the link access procedure for reliable data exchange over a link between a DTE and a DCE.

full-duplex A mode of operation which permits transmission in both directions simultaneously.

gateway access protocol The protocol used between a host DNA system and a DNA system that is a DTE on a PSDN, to provide the X.25 gateway access facility to a user on the host.

general topology subnetwork A non-broadcast subnetwork.

global name That part of the entity name which identifies the node in the network.

global network addressing domain The addressing domain consisting of all NSAP addresses in the OSI environment.

half-duplex A mode of operation which permits transmission in both directions alternately, but not simultaneously.

HDLC High-level Data Link Control. An ISO standard Data Link layer protocol.

header The first part of a protocol message, containing protocol control information to coordinate the operation of the communicating protocol modules.

IDI *See* **Initial Domain Identifier.**

IDP *See* **Initial Domain Part.**

Initial Domain Identifier The part of an NSAP address which identifies the authority responsible for the assignment of the DSP.

Initial Domain Part The part of an NSAP address assigned by the first level addressing authority.

intermediate system A system which relays NPDUs between other systems.

LAN Local Area Network, a communication facility that provides high speed communications in a moderate size geographical area.

layer A grouping of related communications functions which provide a well-defined service to a client, independently of the protocols and other means used to provide it.

level 1 router A router which performs routing within a single area. Messages for destinations in other areas are routed to the nearest Level 2 router.

level 2 router A router which acts as a Level 1 router within its own area, but in addition routes messages between areas.

line A physical path which provides direct communication among some number of stations.

line sharing A form of X.21 switched line sharing in which many clients have access to a line, but only one client has access to any single call.

link state PDU A PDU used by the routing algorithm to exchange information about each system's neighbors.

local name That part of the entity name which identifies an entity within a node.

logical channel An association between a DTE and a DCE for a given virtual circuit.

LSP *See* **link state PDU.**

Maintenance Mode In DDCMP, the mode of operation used by MOP.

Maintenance Operations Protocol A management protocol used for low-level communication with a system which is not fully operational or which is being tested.

management event notification protocol An Application layer management protocol used in DNA Phase V for communication between an event dispatcher and an event sink.

Management Information Control and Exchange protocol An Application layer management protocol used in DNA Phase V.

management model *See* **distributed system management model.**

management protocol The management protocol used between a director and an entity.

MEN protocol *See* **Management Event Notification protocol.**

MICE protocol *See* **Management Information Control and Exchange protocol.**

Modem Connect The name used in DNA for that class of communications links governed by industry standards for modem connection.

module An architectural component of a system which implements a particular protocol, and provides the corresponding service, within the system.

module entity The entity in the management hierarchy corresponding to a module. A module entity is a child of the node entity.

MOP *See* **Maintenance Operations Protocol.**

multicast Transmission of a single frame to multiple destination stations.

multiplexing The operation of multiple layer n connections over a single layer $n-1$ connection. For example, the operation of multiple Transport Connections over a single Network Connection.

multipoint link A communications link connecting more than two stations, where one station is responsible for data link control. Also known as multidrop.

nameserver A system with at least one active clearinghouse.

namespace A tree of directories, starting at a root directory.

NCL *See* **Network Control Language.**

neighbor A system reachable by traversal of a single subnetwork.

network A collection of systems interconnected by lines.

Network Control Language The command line interface to directors.

network management Functions which permit the operation of a network to be controlled and monitored.

network manager A person using network management.

Network Service Access Point An addressable point at which the Network Service is made available.

Network Services Protocol A protocol operating in the DNA Transport layer.

nickname A locally assigned name used to refer to a global name.

node entity The top-level entity in the management hierarchy of a system.

normal mode An HDLC operational mode used in DNA over half duplex links.

NPDU A Network layer Protocol Data Unit.

NSAP *See* **Network Service Access Point.**

NSAP address The address of a Network Service Access Point.

NSAP selector The last byte of an NSAP address, which selects a particular NSAP, and hence a Network layer user, within a system identified by the preceding fields of the address.

NSP *See* **Network Services Protocol.**

null modem A simple form of modem connection where only the data interchange circuits, and not the modem control circuits, are used.

object entry A Naming Service entry which contains the attributes of a network object.

operational mode In HDLC, the particular operational state or protocol being used.

packet level Level 3 of the CCITT X.25 Recommendation which defines the packet format and control procedures for the exchange of packets over a PSDN.

PCI *See* **protocol control information.**

PDU *See* **protocol data unit.**

permanent virtual circuit A virtual circuit that is a permanent association between two DTEs.

physical connection The Physical layer communications path between two systems.

Physical layer The layer of DNA which is concerned with the transmission and reception of data on the transmission medium.

point-to-point link A communications link connecting two stations.

preamble A bit pattern used by the Physical layer to synchronize to bit boundaries, transmitted at the start of each frame.

primitive name A name that denotes a single, unique object.

protocol control information Information sent between communicating protocol modules to coordinate their operation, as distinct from user data.

protocol data unit A message sent from one protocol module to its partner, containing protocol control information and (often) user data.

protocol ID A five byte field in the header of a SNAP frame on a LAN, used to identify the Data Link client at the receiving system that is to receive this frame.

protocol identifier A character string name for a protocol.

protocol sequence An ordered list of Protocol Identifiers.

protocol type A two byte field in the header of an Ethernet frame, used to identify the Data Link client at the receiving system that is to receive this frame.

public data network A data network administered by a public service provider.

PVC *See* **permanent virtual circuit.**

read-only replica A Replica which responds only to lookup requests.

reassembly The process of reconstructing a complete user data message from the received segments.

reassignment When operating over the CONS, the process of transferring a Transport Connection to use a new Network Connection when the original has failed for some reason.

redirect NPDU An NPDU issued by a router when it forwards a data NPDU onto the same subnetwork from which it was received. It includes the subnetwork address to which the NPDU was forwarded. This indicates to the sender of the original data NPDU that it can send subsequent NPDUs destined for the same NSAP address directly to the indicated subnetwork address.

replica A copy of a directory stored in a particular clearinghouse.

retransmission The procedure of transmitting a message for a second or subsequent time. Performed when it is assumed that the previous copy of the message was not successfully delivered.

root directory The top level directory which establishes the root of a namespace.

round-trip delay When operating a protocol using positive acknowledgments, this term is used to describe the total time taken for a message to be transmitted, arrive at its destination, its corresponding acknowledgment to be sent and subsequently received by the sender of the original message.

router *See* **intermediate system.**

routing domain A collection of End Systems, Intermediate Systems and Subnetworks which operate according to the same routing procedures and which is wholly contained within a single Administrative Domain.

SDU *See* **service data unit.**

segmentation The process of breaking a large user data message into multiple, smaller, messages for transmission.

sequence number A field carried in a PDU which indicates the sequential ordering of successive PDUs.

service A set of functions provided by a layer to its client.

service data unit A distinct unit of data passed by a client of a service, for transmission to the remote client; a message.

service element That part of an entity which performs the primary functions of the entity.

Session Control The DNA module providing functions for system-dependent process-to-process communication, name to address mapping, and protocol selection.

SNAP Subnetwork Access Protocol, a form of LLC frame used on LANs where multiplexing is done using a five byte Protocol ID field. This allows higher layer protocols that are not national or international standards to be addressed.

SNDCF *See* **Subnetwork Dependent Convergence Function.**

soft link An alternate name for an object or directory in a namespace. Soft Links allow users to view names as forming an acyclic directed graph rather than a pure tree.

spanning tree As used in this document, a logical topology used for Bridge forwarding in an Extended LAN; it includes all Bridges in the Extended LAN but has no loops.

SSAP Source Service Access Point, the one byte field in an LLC frame on a LAN that identifies the sending Data Link client protocol.

static routing The use of manually entered routing information to determine routes. This is used for routing between routing domains where it is not possible, or not desirable, to exchange the routing information necessary for the dynamic determining of routes.

station A termination on a communications link comprising Physical and Data Link layer entities.

subaddress An additional piece of addressing information for use by a DTE, beyond that needed to identify a particular subscriber line or group of lines.

subchannel A logical communications path within an NSP Transport Connection that handles a defined category of NSP data messages.

subnetwork A collection of equipment and physical media which forms an autonomous whole and which can be used to interconnect systems for the purposes of communication. For example, an X.25 packet switched network, an HDLC datalink, or an ISO 8802-3 LAN.

Subnetwork Dependent Convergence Function The set of functions required to enhance the service provided by a particular subnetwork to that assumed by the Connectionless-mode Network Protocol (ISO 8473).

subscriber line The physical line between a DCE and the local exchange in a public data network.

switched virtual circuit A temporary association between two DTEs.

synchronous link A communications link using synchronous transmission, in which transmission of a block of data is preceded by a special synchronization sequence. Data is transmitted at a fixed rate with the transmitter and receiver synchronized.

system An implementation of a computer that supports the Network Layer, the Transport Layer, and the Session Control Layer. Each system has a unique Network layer address.

template A named collection of module-specific parameters which can be referenced by a client of the module without knowing their individual significance.

ThinWire A system which uses a type of (thin) coaxial cable for use in Ethernet, primarily for use in office area wiring.

topology The logical arrangement of a set of interconnected systems, irrespective of their physical locations.

tower A Protocol Sequence along with associated address and protocol-specific information.

TPDU A Transport layer Protocol Data Unit.

Transport Connection A virtual connection at the Transport layer between two Transport service users.

Transport layer A DNA layer which provides a reliable end-to-end data transfer service between communicating systems. It operates using the service provided by the Network layer.

Transport service user A user of the service provided by the DNA Transport layer. For example, DNA Session Control.

tributary station A non-control station in DDCMP multipoint configurations.

UI frame An unnumbered information frame in HDLC, used to carry data which is not subject to flow control or error recovery.

Videotex A computer based electronic retrieval system which uses a hierarchical arrangement of information.

X.21 A Recommendation of CCITT defining access to circuit switched public data networks.

X.25 gateway access The facility that allows a user in any DNA system to access X.25 packet level services.

XID frame A frame in HDLC used to exchange operational parameters between the participating stations.

Index

Index