XEROX

Xerox Network Systems Architecture

General Information Manual

XEROX NETWORK SYSTEMS ARCHITECTURE GENERAL INFORMATION MANUAL

XEROX

XNSG 068504
April 1985

Printed in U.S. A.

Text and graphics prepared on the Xerox 8010 Information System, and printed on a Xerox laser printer using the Interpress electronic printing standard.

Xerox Corporation
Office Systems Division
2300 Geng Road
Palo Alto, California 94303

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

For nearly three decades Xerox Corporation has played an important role in improving the productivity of office workers. In addition to its widely-recognized line of plain paper copiers, Xerox is manufacturing and marketing word processors, facsimile transceivers, data terminals, electronic typewriters, professional workstations, electronic printers, and a host of related products. These advanced products are being used throughout the world to improve personal efficiency and productivity in handling physical and electronic documents in the office.

Productivity through systems

The size and vitality of the office systems industry is a direct result of the substantial gains in productivity users have been able to realize through modern computing and communication technology. As a result of those gains, work can flow more smoothly, data can be processed more accurately and quickly, and information can be distributed more efficiently. In turn, management can be more effective and responsive in its decision-making.

The early office machines were basically "point products," which means that their benefits were derived essentially from functions wholly performed within the product itself. For example, a point product such as a word processor made it possible for a secretary to be much more productive in many routine tasks, without requiring that the product be connected to anything else. In this fashion many of the more obvious tasks in offices were automated and significant gains in productivity were realized. Xerox was part of this initial wave of office automation with its word processing systems. Moreover, many of the most important subsequent developments in office automation have come from basic research work carried out at the Xerox Palo Alto Research Center.

Productivity through networks

One result of that research was a recognition that further major productivity gains would be made when point products could easily communicate and share work with each other. These new requirements for intercommunication and integration led Xerox to announce, with its partners, Intel Corporation and Digital Equipment Corporation, that Xerox' proprietary local area networking technology, Ethernet, would be made available to the business and academic communities. Xerox' intent was to promote widespread acceptance and use of local area network technology to provide a basic, general-purpose data transmission "highway" within a facility such as an office building, laboratory, or factory.

The importance of local area networks in such applications as office automation cannot be overemphasized. They are one of the most important means by which individual devices—workstations, word processors, electronic typewriters, filing and printing subsystems, etc.—communicate directly with each other. Being able to intercommunicate is a necessary first step toward the functional *integration* of these devices: the coordinated cross-functioning between products that will deliver the next wave of effectiveness and efficiency improvements in offices and related environments. (Note, however, that a local area network, or any other communication system, by itself does not guarantee integration; more is required as we will see.)

Productivity through integration

As an example of functional integration, when the Ethernet announcement was made, Xerox introduced a series of integrated office products called the 8000 Network System, which included the Star Information System along with a number of shared-resource "servers." Star pioneered such concepts as "bit-mapped" (or "all points addressable") displays, "windows" for simultaneously interacting with multiple processes, and a "mouse" for efficient operator control. The 8000 Network System provided not only the functions that were available with the older point products, but an entirely new set of functions such as coordinated document creation, centralized printing, and filing. These functions were possible only because of the high degree of integration among the members of the 8000 family.

More recently Xerox has broadened the integration of its product line, continuing to offer additional productivity improvements. Gradually drawn into the integrated community were such products as word processors, personal computers, very fast electronic printers, an input scanner, electronic typewriters, and special systems for electronic engineers. In each of these cases, integration means that all products work effectively with other products, exchanging data, sharing resources, and building applications, thereby leading to new levels of productivity.

Commitment to open systems

Xerox' development philosophy has emphasized taking maximum advantage of what has been developed before and of the efforts in the international standards groups that are leading to robust, durable standards. The openness of the Xerox approach works both ways: as new developments have matured at Xerox, a conscientious effort has been made to share these developments with the rest of the information systems world. Xerox is committed to an open systems approach to the development of its information systems.

Xerox Network Systems: the key to integration

Although the Ethernet local area networking technology makes it possible for intelligent devices to intercommunicate efficiently, simply having that facility does not ensure the devices will be integrated. All it guarantees is that they will be able to exchange data back and forth. Simply put, the benefits of an integrated information system can be obtained only when the individual elements work together. That, in turn, depends on what those elements do with the data they exchange.

Network architecture

The rules governing the exchange of information among networked devices, and specifying the processes through which work on that information is to be done, are collectively known as a network architecture. There are many different types of network architectures, varying with respect to their application orientation, the emphasis placed on local vs. wide area communications, etc. But they all define a structured approach to the exchange and handling of information in a network, and they all encompass greater functional scope than the comparatively straightforward matter of data transmission provided by the network itself. In fact, a general purpose network architecture is capable of employing a number of different transmission techniques as required: a local network here, a wide area network there, along with satellite links, public data networks, etc.

An analogy might be drawn between network architectures and city planning. At first glance, a "city plan" could be thought of as a map of the city's streets. But in order to have a completely integrated city plan, provisions must be made for various city services, the delivery of utilities, the extent of land use restrictions, the establishment of public activities such as education and recreation, and the integration of various transportation systems, of which the city street system is an important part, but only a part. The city streets are analogous to a local area network and the overall city plan is analogous to the network architecture.

Long-range planning

As is the case with a comprehensive city plan, a good network architecture is conceived with the understanding that not all eventual uses of the architecture can be anticipated in advance. But it accommodates known elements successfully, is faithful to long-range goals, and provides for future growth and change in ways that do not require the architects periodically to return to their drawing boards to start over.

The network architecture underlying Xerox' focus on integrated systems is called Xerox Network Systems (XNS).

XNS provides a conceptual framework for accomplishing all the functions required in a general-purpose information system.

Like the city plan that uses city streets as one of its building blocks, XNS uses Ethernet (along with other networking techniques); but XNS is not synonymous with Ethernet. Indeed, XNS imposes extensive and stringent requirements on those products designed according to its rules, requirements far more elaborate than those applicable to devices that simply connect to Ethernet.

Integrated office systems

Following the rules and specifications of XNS, it is possible to design highly integrated information systems, using hardware and software elements designed by different groups using different technologies. XNS brings to this process a broad range of facilities and functions, tested in a variety of implementations over a long development period. This makes XNS one of the most thorough and robust of the extant network architectures, particularly among those intended for commercial, office-oriented applications.

XNS is one of the major reasons why Xerox products are as capable, reliable, user-friendly, and obsolescence-resistant as they are. These qualities are specific objectives of the architecture design and reflect directly on the products that the architecture supports. Office products not designed within a framework such as XNS provides will likely be lacking in one or more of these important attributes.

Network architectures and distributed systems

A network architecture provides the conceptual framework for the design of the many functions that are necessary for network elements to work together. It also provides a series of specifications for the common functions that must be agreed upon by the network community. These functions usually take the form of protocols that provide for accomplishing specific tasks such as the transfer or storage of information. In such instances the network architecture can be thought of informally as a collection of protocols, each of which is usually identified with the service to which it corresponds.

Traditional architectures are centralized

In addition to XNS, a number of other network architectures have emerged within the computer and office equipment industries. Two examples are IBM's Systems Network Architecture, which is oriented to the interconnection of mainframe computers and terminals, and Digital Equipment Corporation's Digital Network Architecture, which supports the interconnection of that company's small- to medium-sized computers.

An important distinction exists between XNS and the typical computer-manufacturer network architecture. To one extent or another, the latter tend to be hierarchical in organization, intended for applications where one or more computers dominate the resulting information system and its users. In such systems other elements—satellite processors, terminals, workstations, etc.—are clearly subordinate to the large computers. In various ways the network architectures underlying these systems are designed to create and reinforce this relationship.

There is a place for such arrangements. Traditional mainframe computing is usually organized in such a way that information naturally flows to and from major computing centers. The operation of other system elements is intended to support the processes taking place in the centralized machines. In many respects this view of computing reflects traditional data processing implementation.

Centralized systems lack flexibility

Many networking applications, however, are not well served by this model. In particular, many of the processes and activities in modern offices are essentially autonomous, i.e., initiated and conducted by individuals at their own pace, with only occasional references to external resources. This is particularly true with the creation, editing, storage, retrieval, and printing of documents, the common currency of the office. In most cases document management requires dealing with a large series of autonomous processes, for which the centralized mainframe-oriented model of information flow and processing is often not very effective and a potential bottleneck.

XNS architecture is distributed

For this reason XNS—and network architectures similar to it—are designed to support autonomous processes, implemented by distributed, rather than centralized, processors. One of the important results of this is that XNS makes it possible for networks to grow incrementally through replication of the individual system elements and data bases. XNS is a distributed/replicative network architecture.

A more complete discussion of network architectures and the relationship of XNS to other architectures and industry standards is provided in the Appendices.

Realization of Xerox Network Systems

This manual describes the architecture of Xerox Network Systems. It provides information on the standards and protocols that comprise the architecture. Detail specifications and specific hardware and software products are not described, but a description of the services is provided to illustrate how the protocols are used and how the network architecture integrates products to form systems. Fig. 1-1 shows the relationship between the XNS architecture, standards, and products.

The architecture is at the top of the hierarchy. It establishes the general structure and functioning of the network. The specifics

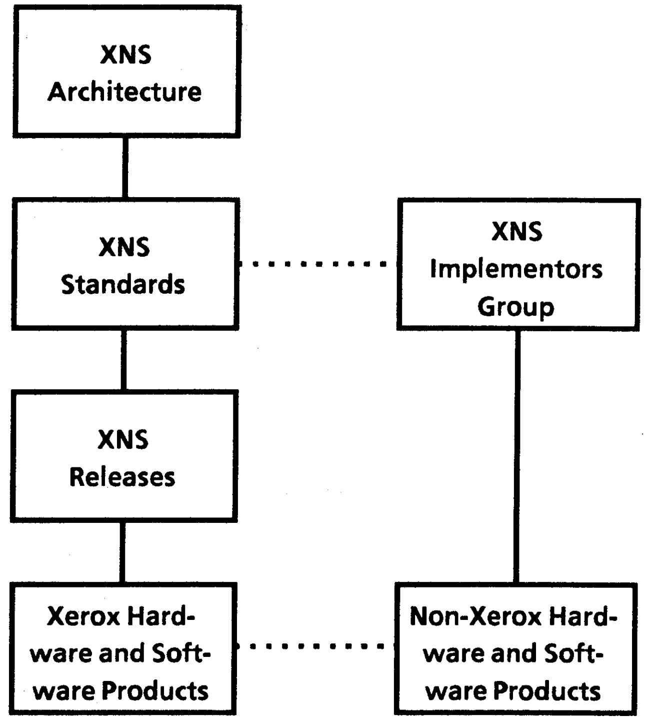


Figure 1-1 **Structural relationship of architecture, standards, and products**

of the architecture are contained in the various standards documents. The Xerox Network Systems releases specify which Xerox hardware and software products conform to the standards, and their level of mutual integration.

XNS architecture is open

There is a well-defined path for other companies who choose to offer products in conformance with XNS standards (see Fig. 1-1). Xerox offers a variety of help to companies wishing to adopt any or all of the XNS standards. This help includes publication of standards and guides, assistance in implementation through the XNS Implementors Group and the XNS Institute, and a variety of joint intercompany arrangements. For further information please contact:

Xerox Corporation
Xerox Network Systems Institute
2300 Geng Road
Palo Alto, Ca 94303
(415)-496-6088

Xerox' goal in office automation

Xerox' goal in office automation is to provide its customers with products, systems, and services to maximize return on their information assets. Information assets come in two forms: people and hardware/software products. Increasing the return on information worker assets is increasing effectiveness; increasing the return on information product assets is increasing efficiency. Fig. 2-1 graphically displays the concept of return-on-information assets (ROIA). The boxes at the bottom of the diagram show some of the ways in which the return can be increased and the assets decreased.

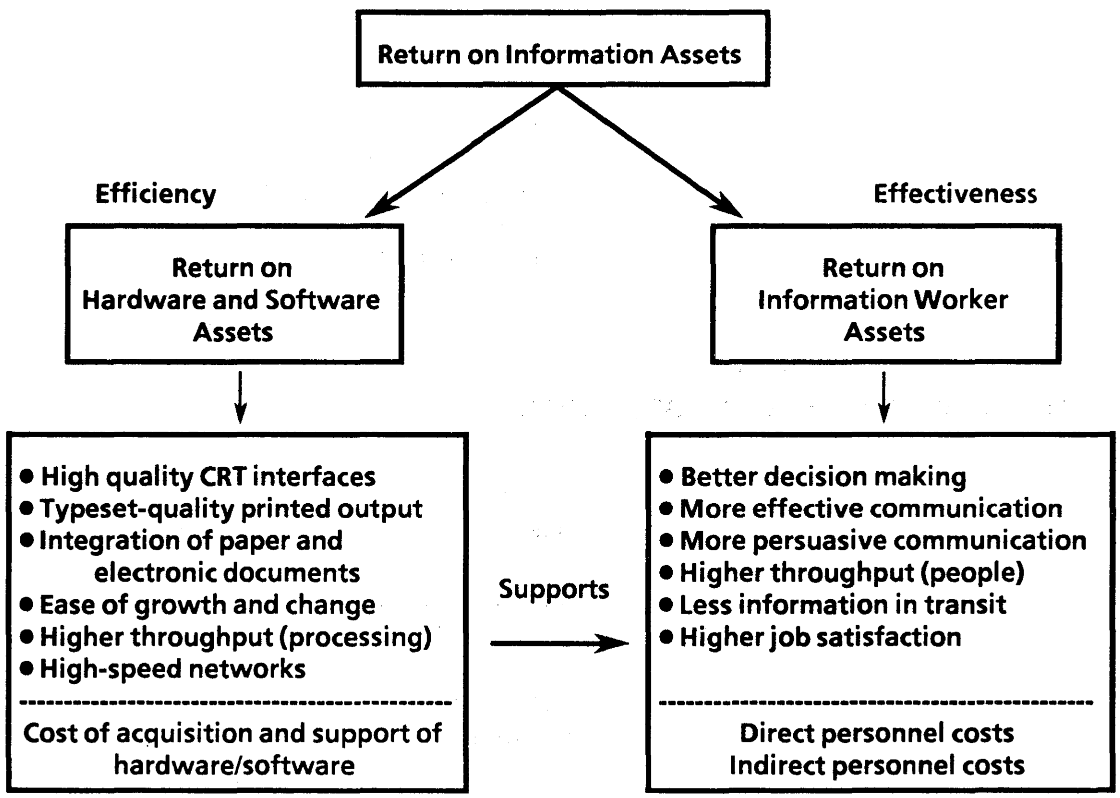


Figure 2-1 **Return on information assets**

Typeset quality

Consider an electronic printer which can provide typeset quality directly. Not only can documents be printed on fewer pieces of paper, thereby reducing material, storage, and delivery costs, but higher-quality documents can support more effective communication and decision making.

People come together to accomplish information related tasks in offices because these tasks are generally highly interactive. The means of interaction is usually either through verbal means, including physical cues, or through the creation and reading of documents.

Documents and document management

A clear understanding of the concept of a document is essential to understanding the objectives of XNS. A document is a structured organization of information designed to communicate effectively with people. A document may be rendered by a printer, which produces a paper document from an electronic original, or it may be rendered by a workstation, which makes an electronic document visible. Alternate forms of documents include voice documents and video records. In the future it will be possible to integrate these various document forms; for example, a text document may be annotated by digitized voice.

As most documents are stored at one time or another, it is easy to confuse documents with stored information. Not all stored information is in document form. A computer data base, for example, is not a document since it is organized to provide efficient storage and access to elements of information, not to communicate with people.

The general objectives of XNS is therefore to increase the ROIA by facilitating the creation, capture, storage, communication, printing, and replicating of electronic or paper documents within the office, especially at the work group and departmental levels. This is what Xerox calls document management.

Xerox Network Systems objectives

Systems that fulfill specific short-range objectives are always possible, and may appear to have a lower initial cost. The real test of a system's quality, however, is if it can be used over a long period as applications change and new technology is introduced. In the long run, systems that lack proper architectural support are seldom the most cost effective. They become obsolete in a short time and have to be replaced, often resulting in costly disruption for users. Xerox Network Systems Architecture, with its long-range view, is for those users and suppliers who want to do it right the first time.

The XNS objectives include:

Formal definition of standards

The creators of XNS knew that this architecture would be used as a basis for the design of a wide variety of products. The architecture, therefore, would have to be powerful and open-ended. The products supported by the architecture would be designed by a number of Xerox and non-Xerox organizations.

Some of these organizations would be as distantly removed from the group responsible for the architecture as if they were outside of Xerox. For this reason, the architecture had to be formally defined and its individual elements subject to strict rules of standardization and configuration control.

Document compatibility

One of the key issues in the design of an information system is the preservation of compatibility for the users' work products (documents of all kinds). Once a document is created, it must be capable of being operated upon at some point in the future and at another part of the system (even in another part of the world where a different language is used). If at all possible, architectural changes that "dead-end" already-created documents must be avoided.

Product performance

The architecture must be designed so the products it supports can be engineered in accordance with it. Using proven technology, the products must still meet their performance, reliability, and cost goals. This objective separates the theoretical from the practical as each architectural provision must be implemented with real hardware and software at a competitive cost. An appreciation of this must influence architectural design work.

Product evolution

The architecture must make it possible for products and services to evolve in two directions: toward greater breadth (new areas of applications, user categories, operating environments, etc.), and toward greater depth (new functions).

Interconnection with other systems

Xerox systems must interface and interact with computer products and specialized systems from a large number of suppliers. Because Xerox' autonomous distributed/replicative system must be closely integrated with hierarchical, mainframe-oriented systems, some interfaces become very complex and create profound challenges for the architectural designer. Despite the difficulty, these challenges have been met. The end user and the industry are not well served by approaches that omit interconnection with external mainframe systems.

Industry standards

Xerox has a strong commitment to the development and use of industry standards. Xerox personnel have participated actively in standardization efforts sponsored by groups such as IEEE, ANSI, ISO, ECMA, and CCITT. XNS architecture will either adopt or be compatible with all the important relevant standards. Many aspects of XNS have yet to be the subject of external standardization since in many cases Xerox' work on XNS tends to lead official standards formulation and adoption processes by several years. Nevertheless, Xerox anticipates making XNS-derived contributions to such efforts when they are finally undertaken just as it has with past and present standards projects. A summary of the XNS relationship with key international standards appears in appendix C.

Qualities of Xerox Network Systems

The layers and functions of XNS bring together an important set of qualities on behalf of the products they support. Together these qualities ensure that a system designed with the support of XNS will be among the most powerful, cost-effective, and obsolescence-resistant systems available.

The XNS qualities include:

Maturity

XNS is more than just a theoretical construct; Xerox has been working on it for over a decade. Thousands of man-years of effort have gone into its design, implementation, testing, and refinement. It is a practical system with well-established performance and functional characteristics. XNS is a key support element in Xerox' distributed information systems.

Distributive/replicative

Xerox has designed XNS to support distributed processing in which autonomous devices are interconnected in a network that permits simple, low-cost, incremental expansion. Because it is easy to replicate workstations, file and print servers, and other network resources as a user's needs grow, a user will be able to benefit from the system for a long time—unlike mainframe-centered architectures.

Completeness

XNS is one of the most complete architectures available. It provides two separate information exchange techniques: one for moving data and the other for invoking (and responding to) remote processes. It also provides a broad spectrum of self-contained application processes. Including these processes within the architecture rather than leaving them for user development means that an XNS-based system is more immediately useful to the user and ensures uniformity and robustness in all the important network functions.

Growth and expandability

XNS is not a static architecture. One of its strengths is that new facilities can continue to be added without major disruption to the product hardware and software it supports.

Transparency to the user

Despite its many functions and its potentially worldwide scope, XNS is transparent to the user. Information is exchanged, resources are added to and subtracted from the network, remote procedures are invoked, and all activities defined by the architecture take place while the user remains free to do his job without concern for what is making that job possible.

Global scope

Whether the physical scope of a user's network is limited to a single building or whether it spans the world, XNS provides a single, consistent set of services that automatically adjust to the network's scope. Each user sees one logical network, even though that network may consist of many physical local area networks interconnected by a web of wide area telecommunication circuits.

An "open" system

XNS systems are multivendor systems. As new, specialized hardware and software are introduced for use in office applications, these facilities will be capable of integration within an XNS-based system. Xerox began this process in 1981 with the public disclosure of the non-application layers of the XNS architecture. The application layers are also being fully disclosed as individual applications reach technical maturity.

Xerox Network Systems concepts and facilities

The basic idea behind most network architectures, including XNS, is that of a layered structure. This means that the various functions supported by the architecture are divided into a series of layers. By convention, the most primitive tasks are located in the lowest numbered layers while the higher layers are reserved for more sophisticated tasks.

ISO Open Systems Interconnection Reference Model

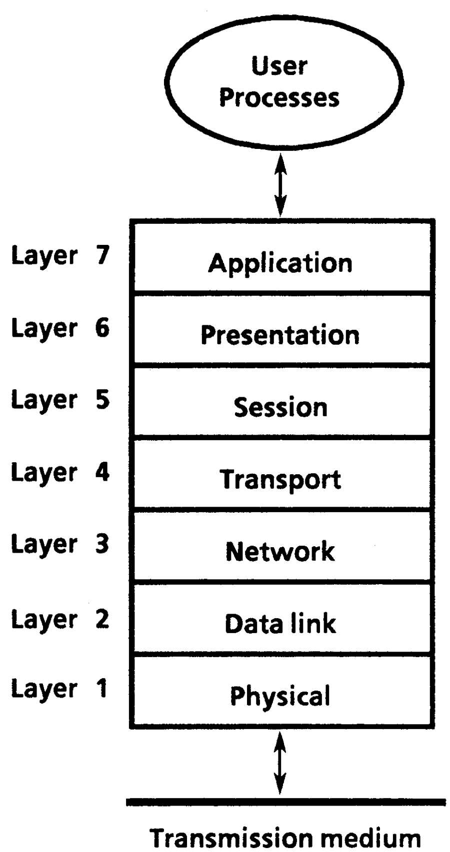


Figure 2-2 **Layers in the ISO Model**

A very useful model of this layered structure was developed by the International Organization for Standardization (ISO). The model is called the "Open Systems Interconnection (OSI) Reference Model," referred to as the "ISO Model" in this document. It was adopted in 1981 under the sponsorship of ISO's subcommittee 16. One measure of the great need for this clarifying set of definitions has been the widespread, adoption of the ISO Model concepts, even among people who are not professional network architects. Although lacking sufficient detail to be a standard, the ISO Model provides a common basis for discussing a complicated subject.

Figure 2-2 shows the essence of the ISO Model, which holds that all network architectures, no matter how complex or broad-guaged, can be divided into seven functional layers. Each layer is superior to the one below it, and subordinate to the one above it, in the sense of the relative primitiveness or sophistication of its function. This sense of hierarchy is an important feature of the ISO Model; a commonly-accepted perspective is that the functions of one layer use the resources of the lower layers to complete their assigned tasks. Of course, the entire hierarchy exists in the final analysis to serve a set of user processes. Because of the hierarchical nature of this model, the lower layers typically deal with ordinary data communication matters, while the upper layers deal with broader issues of information system control, management, and applications.

The individual layers in the ISO Model, and the roles they play in a network architecture, include:

Layer 1: Physical

All network architectures provide for data transmission. In this layer the fundamental tasks related to transmission take place, such as bit stream manipulations, dialing (for switched networks), modem control, etc.

Layer 2: Data Link

This layer is basically responsible for getting information reliably across a data link. Information is organized for purposes of transmission (outgoing) and reorganized for processing (incoming); transmission errors are detected (and sometimes corrected), and the rate of data flow through the link is controlled.

Layer 3: Network

Fundamentally this layer is responsible for getting units of information across the network. It provides data organization functions for information moving on and off the transmission media; it also includes higher level forms of error recovery, and most of the control of network addressing, routing and switching decisions.

Layer 4: Transport

In this layer decisions are made concerning which transmission service is appropriate, more data organization and reorganization functions are provided, and a set of network management tasks are performed.

Layer 5: Session

This layer is responsible for creating a communicating relationship between two parties for the time necessary to complete their interaction (a "session"). Related tasks such as buffering and queuing are also performed within this layer.

Layer 6: Presentation

Data conversion and similar tasks are performed here. The object is to translate data forms from those understood by user applications into those used in lower layers (or remote user processes), and vice versa. This can range from simple matters of code and format conversion, to complex matters of syntax changes (as in the extreme case of shifting between two different programming languages).

Layer 7: Application

Specific applications are performed here; for example, in a document-oriented environment, this layer might include filing, printing, mailing, etc.

The ISO Model also helps to explain interfaces and protocols. Fig. 2-3 shows the ISO Model expanded to include two separate nodes. (A node is any logically or physically distinguishable entity on the network, such as a terminal, a computer, or a program running inside a computer.) The ISO Model assumes that the two user processes have need to communicate (e.g., one user wishes to transfer a spreadsheet file to another user), or that one user process has need to use resources in the other node (e.g., the user wishes to print a document on a remote printer).

Interface between layers

A formal interface is conceived to exist between each layer of the ISO Model. That interface is the "logical view" from a higher layer to the next lower layer, and vice versa. Information related to actions wanted and results obtained, as well as actual data, is passed back and forth across these interfaces.

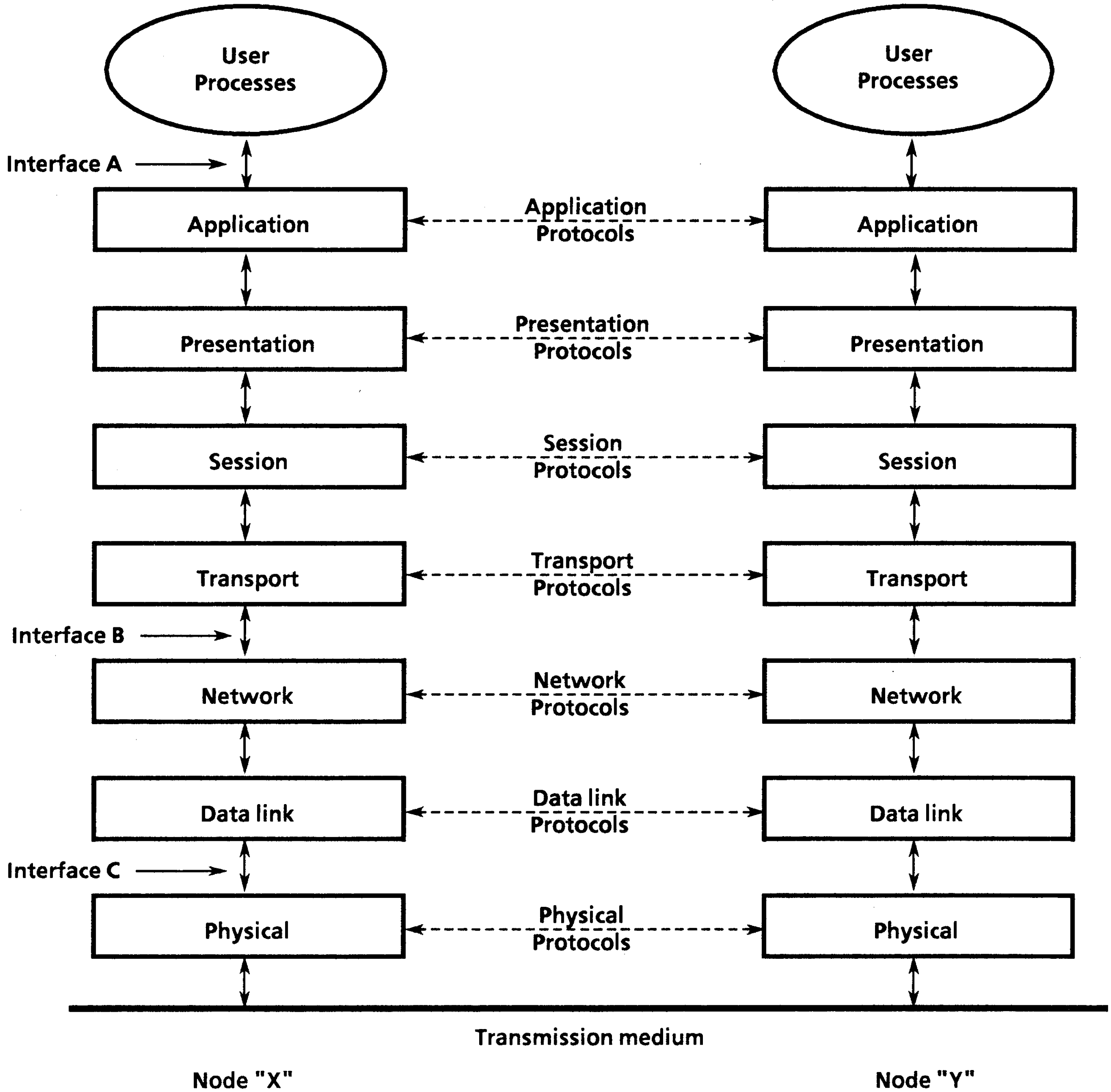


Figure 2-3 **Protocols and interfaces in the ISO Model**

A number of the ISO Model interfaces have counterparts in real networking systems. In Fig. 2-3, Interface A is the interface the user processes see to the entire network architecture. An example of Interface B is the widely discussed X.25 packet switching interface. An example of Interface C is the popular EIA RS-232-C data communication interface.

Protocols for each layer

When communication is taking place between two nodes, the work done between these nodes is conceived to be structured into a set of protocols. A protocol is an organized set of rules for getting work done. In the ISO Model each layer in one node is conceived to be working with the corresponding layer in the other node, by means of peer protocols. In other words, a given protocol represents a dialogue between two equally potent functions, each of which operates through its upper and lower interfaces to perform its specific tasks.

The ISO Model is a "guide," not a "specification." For various practical reasons, real network architectures do not necessarily formally define all the interfaces and protocols suggested by Fig. 2-3. A real architecture might have several different application protocols, no presentation protocols, and so on. Nevertheless, the basic themes of layered structure, interfaces, and protocols are important to XNS, as well as to nearly all network architecture design.

XNS structural overview

Fig. 2-4 shows the basic structure of Xerox Network Systems. As in the ISO Model, the XNS structure is organized into a series of layers, approximately corresponding to the ISO Model layers listed on the left side of the illustration.

XNS layers

The XNS architecture groups some of the ISO Model functions into fewer layers for convenience. Each XNS layer corresponds functionally to the ISO layers. Even when the functions are grouped, as in the case of Ethernet, a separation is maintained between the physical and data link layers to allow different physical media to be used with the same Ethernet data link protocol. The XNS architecture is particularly open-ended in respect to multiple transmission protocols corresponding to different types of communication services, and to multiple application protocols corresponding to different functions performed within the architectural boundaries.

As in the ISO Model, the XNS bottom layer provides for physical transmission interfaces. The functions provided by the XNS architecture rise in hierarchical order—through communication control (Internet) and remote procedures (Courier)—to the application layer. Although the lower layers are important, the full functional richness of XNS is revealed by an examination of the application layer. This layer provides a great variety of important office and computing functions to XNS users.

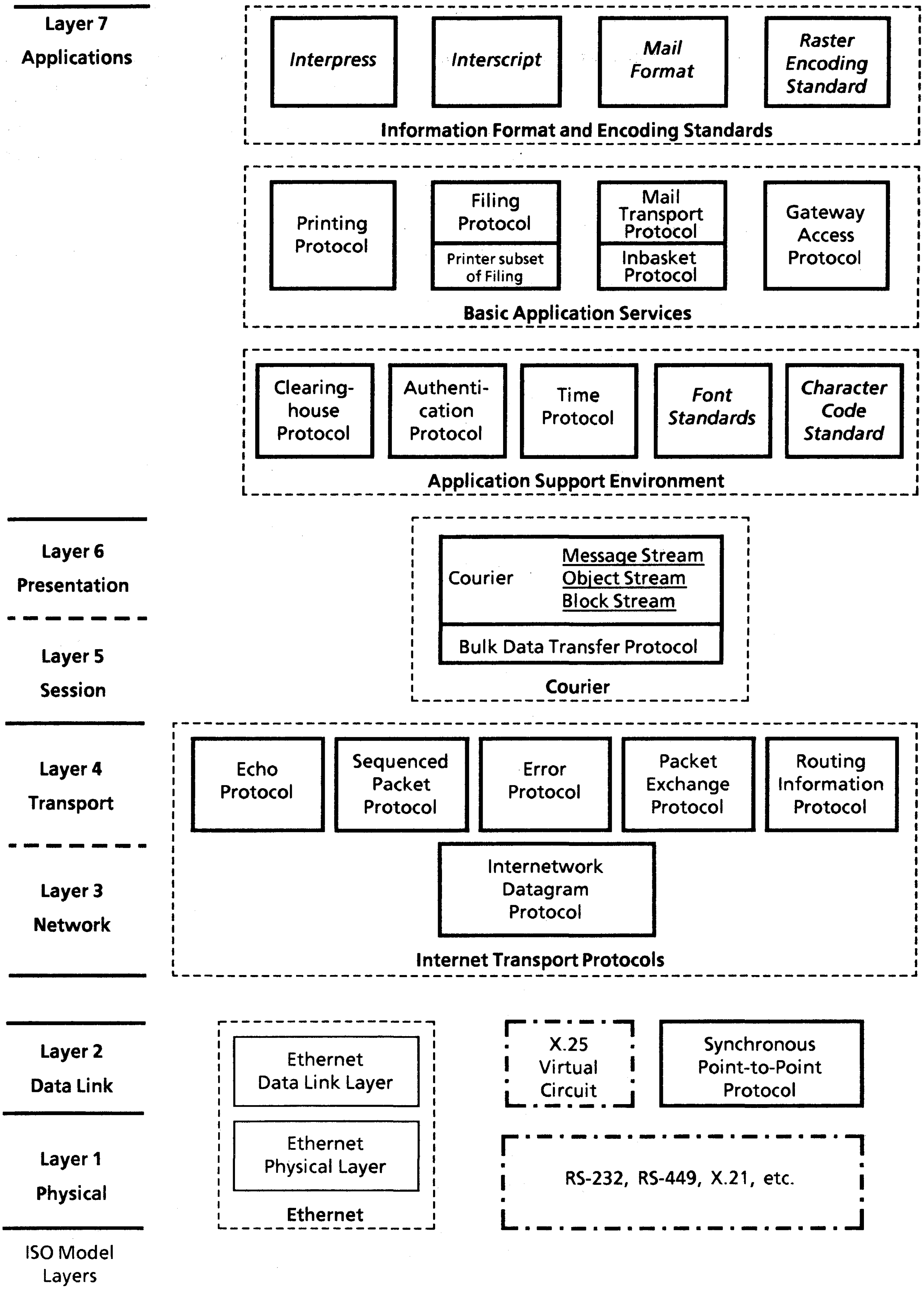


Figure 2-4 **Overview of Xerox Network Systems**

The following explanation applies to Fig. 2-4:

Physical interfaces

At the lowest layer, Ethernet provides its own unique physical interface. It is unlike traditional data communication physical interfaces, which are shown in the box to the right (RS-232-C, RS-449, X.21, etc.). These are shown in a broken outline because, strictly speaking, they are part of XNS by adoption rather than by special design.

Data link protocols

At the next lowest layer, Fig. 2-4 shows the CCITT X.25 Virtual Circuit Protocol in a dashed box to indicate that this protocol is part of XNS by adoption. It is used as part of XNS utilization of packet-switching data networks.

Internet transport protocols

Internet is shown as a set of protocols corresponding to ISO Model layers 3 and 4. The word "internet" is also used to refer to the set of all interconnected Ethernets in different locations, a relationship implemented by these protocols.

Courier

XNS implements the session and presentation layers in Courier, the XNS protocol for remote procedure calls [requests].

Application protocols

At the application layer (ISO Model layer 7) the Application Support Environment provides support resources called on by users and/or the application protocols shown immediately above. These protocols—mailing, printing, filing, and gateway access—are implemented in hardware/software to provide the XNS application services.

Document formats

Within the application layer, the standards for the format or language for the encoding of document form or content are labeled with italic type. In many respects, the utility of XNS depends as much on the innovative approach to document descriptions as it does on the actual protocols. The document encoding techniques referred to in Fig. 2-4—particularly Interscript and Interpress—make it possible for XNS documents to be edited, printed, or communicated anywhere on the system. Other encoding standards are the Character Code Standard for representing text in many languages, and the Raster Encoding Standard for representing compressed and uncompressed bitmap images.

Integration and compatibility

The internal structure of XNS enables effective integration between individual hardware and software elements within XNS-compatible products. Techniques are also provided within XNS for bi-directional protocol and format conversion, permitting other systems to achieve integration with XNS.

Note

Each architecture element shown in Fig. 2-4 is discussed in the following sections, and appendix D gives examples of protocol usage. Further information is contained in documents listed in the annotated bibliography in appendix F.

Network devices and terminology

The XNS architecture consists of a hierarchy of protocols and related formats and encodings. The different network devices connected to a network can communicate with each other if they use the same protocols. The devices that use XNS protocols and connect to the network are called system elements. They are also known as XNS hosts or XNS citizens. These system elements are generally classified as workstations or servers.

Workstations and terminals

Fig. 2-5 illustrates some key system elements. Workstations are devices with which a human operator directly interacts. The term "workstation" is loosely used in the industry to mean various things. In XNS a "workstation" refers to any system element used directly by a person. This would include the Xerox Star, the Xerox 860 word processor, and Xerox and IBM Personal Computers (when these are connected to the network via XNS protocols).

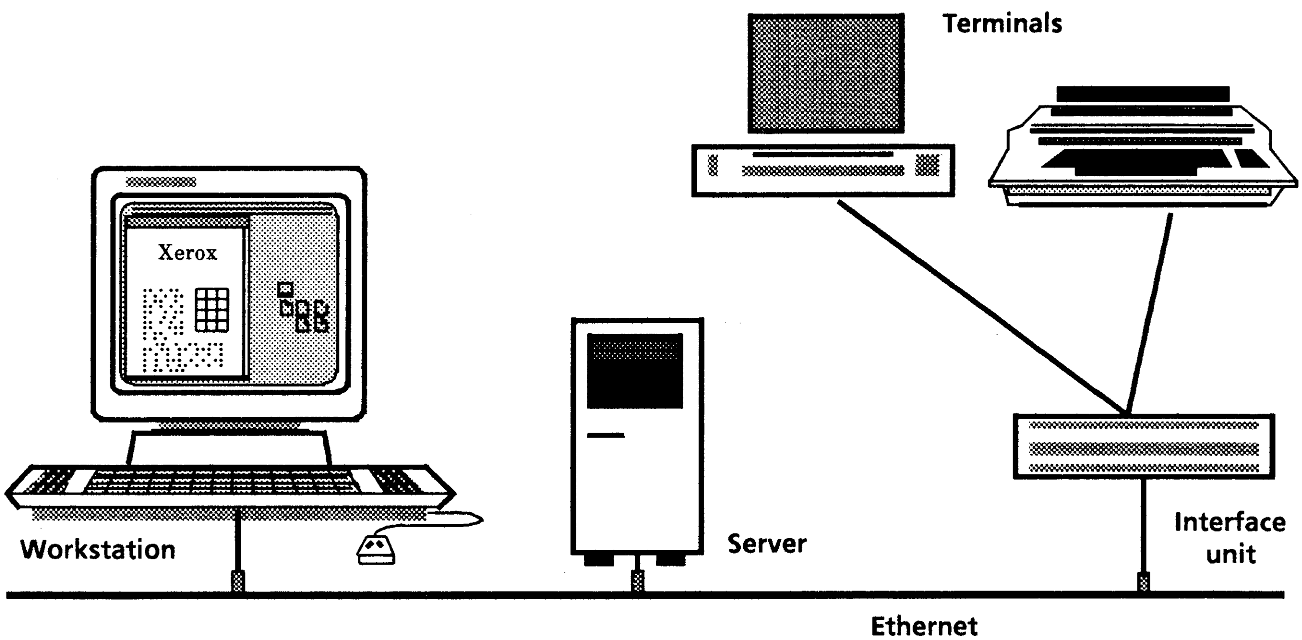


Figure 2-5 **Network devices**

Any product indirectly connected to the network using protocols other than XNS is considered a terminal. Terminals may or may not be intended for direct human interaction; a conventional ASCII keyboard/display unit would be but an ordinary minicomputer may not be. Both would be considered terminals if they were *indirectly* interfaced to the network. Other examples of terminals include the Xerox electronic typewriter or personal computers without a direct XNS connection.

The significance of direct XNS connection is that ordinarily a directly-connected device is expected to implement all the layers of XNS appropriate to its function, which would include *at least* all the layers upward through Courier (see Fig. 2-4), plus selected application protocols. These XNS system elements are assigned a unique host number identification.

This is not necessarily true for terminals which are interfaced to the XNS community by means of interface units such as the Interactive Terminal Service. Through the joint action of the interface unit and the terminal, the XNS protocol rules are obeyed, but the terminal itself is not recognized as a system element and does not have a host number identification.

Servers and services

A server is a device connected to Ethernet whose purpose is to provide a service to network users. The services are high-level functional activities such as filing, printing, mailing, and external communications. They represent high-level applications performed within the architecture. The XNS services implement various application-layer protocols, as depicted in Fig. 2-4. The services are typically collections of software acting *according to the rules of the architecture and its protocols*, to achieve the desired purpose. A server, therefore, is the physical means by which a service is performed.

The word "server" is used in several different ways, depending on the circumstances. Some servers are specifically designed to perform their assigned function, and little else. These dedicated servers tend to consist of the necessary computer peripheral related to the service—a printing subsystem, for example, or a large disk file—coupled to the network by means of an electronic subsystem.

Some servers are actually general-purpose minicomputers programmed to perform the requisite functions; file servers are often implemented this way. Such servers may be capable of accomplishing many other functions than that which makes them servers.

And finally, certain workstations themselves can also function as servers (given the appropriate peripherals and software), in addition to performing their workstation duties—sometimes alternatively, sometimes simultaneously.

One should therefore think of a "server" as any collection of the necessary electronics, software, and peripheral equipment necessary to deliver a service—regardless of what other role that ensemble may also have.

Clients

Clients are the entities which request the performance of a service. Since the XNS services implement application-layer protocols, an appropriate model is that the client uses the application-layer protocol to request the service.

An analogy for the relationship between a client and a service can be found in a human operator using a workstation and possibly other physical network resources to cause some action to take place (e.g., the printing of a document). Since the human cannot directly request the service, he or she works through the user interface of the workstation to cause the desired result. In a similar fashion, client programs use network resources, including other programs, to invoke services.

Services themselves may be clients of other services (e.g., when a filing service asks a Clearinghouse service for information on behalf of a client).

The discussion in this manual focuses on services, and not on the workstations and terminals or their user interface. This is appropriate because workstations are the clients of the described services, and because the user interface to the services varies among the many types of workstations supported in XNS.

User interface

The user interface on Xerox professional workstations provides an easy way for anyone to create, file, mail, or print information. Documents may contain text in many different type styles and sizes, plus charts and drawings integrated with the text. Documents appear as they will be printed ("what you see is what you get," or "WYSIWYG").

Most system options appear on "pop-up" menus and can be activated by merely pointing at an option and clicking a button on a device called the "mouse". Available network resources are shown as pictorial images on the screen (called icons). The icons in Fig. 2-6 show an "in" and "out" basket for mail, different printers, a [remote] file drawer, several documents, and folders containing any number of documents. A user interface like this makes services on the network readily accessible. User actions are intuitive, always consistent, and extremely friendly.

Multilingual capability

The Xerox professional workstations provide many different versions for the user interface, including icons, keyboards, and data formats to accomodate the needs of a worldwide user community.

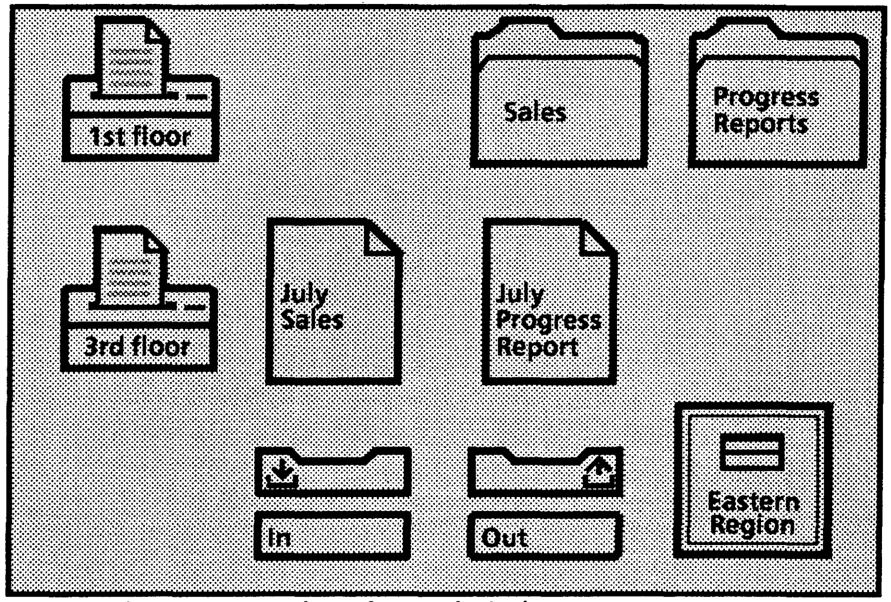


Figure 2-6 **Portion of a user's desktop**

Names

As is the case with most modern network architectures, Xerox Network Systems employs a technique in which certain network resources are "named," that is, identified with labels that relate directly to their characteristics or purpose. This may be thought of as a technique for indirect addressing. The resource is known by its name rather than by its literal or physical address or other location identification. That makes gaining access to the resource substantially easier. The access to system resources will be further explained in the discussion of Clearinghouse. XNS also provides methodologies for naming key system resources (e.g., the fonts used in printing documents).

3. COMMUNICATIONS

The XNS communication facilities are responsible for moving information between network system elements. Owing to the nature of offices (relatively short distance between elements), much of the "traffic" in an XNS network is routinely transmitted across the Ethernet "baseband" local area network. There are other forms of local area networks, however, and XNS is capable of operation over most of them, including multi-channel "broadband" networks. Beyond the relatively short range of such networks, alternative wide-area transmission techniques must be used like conventional telephone circuits (switched or leased), wideband circuits, and public data networks.

Each of these types of service requires specialized data link protocols, such as the Ethernet data link and the Synchronous Point-to-Point protocols. It is one of the strengths of Xerox Network Systems that in most cases new classes of communication services can be accommodated through the addition of new transmission protocol modules ***without disturbing the protocols in the higher layers***.

In addition to basic data link protocols, XNS provides higher level communication control functions embodied in the internet. An implementation of those control functions for purposes of interconnecting remote Ethernets is made by the Internetwork Routing Service which is an integral part of XNS communication facilities.

Ethernet

Ethernet is the local area communication network (or LAN) developed by Xerox, Digital Equipment Corporation, and Intel Corporation. Its specifications determine the kind of transmission medium (coaxial cable), electrical signaling levels, and the data link or transmission protocols.

The Ethernet transmission protocols are independent of the medium. Products supporting these protocols may communicate with each other using fiber optics, twisted pairs of wires, or even radio broadcasting into the "ether." With passage of the IEEE 802.3 standard, Ethernet is now an internationally accepted communication standard. There are some minor differences between the version of Ethernet that Xerox has been using for several years and that adopted recently bv the

IEEE 802 Committee and ISO. Now that this standard has been accepted, Xerox is migrating its products to the official 802.3 version of Ethernet, a process that started in 1985 with the incorporation of standard 802.3 transceivers and controllers into the product line.

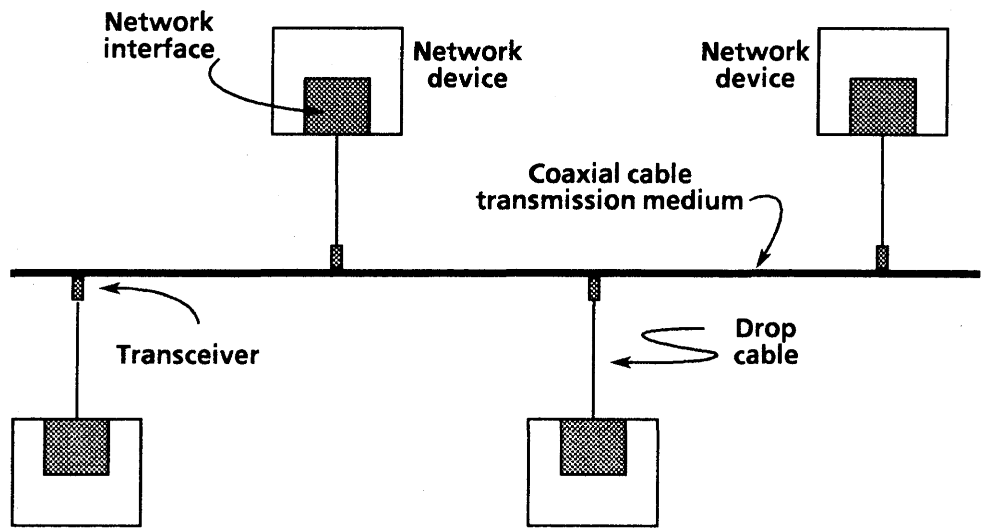


Figure 3-1 **A local area network**

Ethernet components

Fig. 3-1 shows the hardware components required for a baseband Ethernet using coaxial cable. The cable itself is passive; no central resource (power, electronics, computer) is required to allow individual system elements to communicate with each other. System elements attach to the cable via a tap, control electronics (transceiver) located near the cable, an extension drop-cable, and further electronics (network interface controller) located in the system element (network device). The Ethernet cable may be several thousand feet long and, with the use of repeating devices (which strengthen signals on the cable), is normally capable of extending through most facilities.

Network devices may be nearly anything: workstations, terminals, computers, or specialized subsystems of various kinds. The only requirement is that they possess the electrical interface with the cable and the internal logic (usually in the form of software) to interact appropriately with other devices on the cable—in other words, to implement the requirements of the network architecture associated with the network. Other devices (those not possessing the special hardware and software interfaces) can also attach to the networkby means of intermediate adapters.

Ethernet employs coaxial cable as its transmission medium. Data are transferred on the cable at 10,000,000 bits per second (10Mbps), which makes Ethernet one of the highest performance local area networks available. It is the leading example of baseband local area network technology in which computer-type (digital) electrical signaling is applied directly to the medium.

Ethernet benefits

Here are some of the respects in which local area networks differ from conventional data communications:

High-speed communication

Ethernet permits the exchange of data at far greater rates than ordinary telephone-based systems (more than a thousandfold faster!). This enables the development of more sophisticated applications and also permits far greater numbers of devices to be supported over a single cable. (As document complexity grows the need for higher data rates becomes more important. For example, the digitally-encoded form of a 10-page report, containing text, graphics, and pictorial elements, might easily require 2.5 million bits; when a system is in full operation by a number of users, the efficient exchange of bodies of information of this size requires multi-megabit transfer rates.)

Distributed control

Ethernet is designed so that little, if any, equipment is actually associated with the network itself. This is in contrast to a conventional system in which PABXs, central office switches, "front end," and other communication control systems, concentrators, etc., are all employed simply to control communications. In Ethernet, communication control is the responsibility of the attached devices.

Expandability

Ethernet facilitates graceful growth from a network of a few devices to a major network encompassing an entire facility.

Compatibility

Ethernet, together with XNS comunication protocols, enforces compatibility among the attached devices.

Maintainability

Ethernet can be installed and managed by a user rather than an outside vendor (such as the telephone company). This ensures maximum flexibility in the installation configuration and few obstacles when it becomes necessary to modify the network.

Reliability

Ethernet is completely independent of cable amplifiers, frequency converters, and other devices, the failure of which can disrupt a major part of the network.

Fairness

Ethernet accommodates the exchange of information by system elements built by different manufacturers. Any system element conforming to Ethernet rules may tap the cable. The conventions for gaining access to the Ethernet provide fairness among all system elements.

Ethernet architecture

The major division in the Ethernet architecture is between the physical layer and the data link layer, corresponding to the lowest two layers in the ISO Model. The interface between the higher network client layer and the data link layer includes facilities for transmitting and receiving frames, and provides status information. The interface between the data link and the physical layer includes signals for framing (carrier sense, transmit initiation) and contention resolution (collision detect), facility for receiving and transmitting serial bitstreams, and a wait function for timing.

Physical layer

This layer performs all the functions needed to transmit and receive data at the physical level while supporting the data link layer interface. The Ethernet specification describes the 10Mbps baseband coaxial system. Other physical layers have also been specified for broadband channels, fiber optic channels, and twisted pair wires. All these diverse physical media work with the same Ethernet data link protocol. It is Xerox' intention to incorporate other physical media into its product line as they are adopted by the standards organizations and are accepted by users. It is important that there be agreement among vendors and users about the physical media that will be installed for local area networks so equipment from different vendors can operate together. This will avoid costly redundant wiring installations. Xerox has been actively supporting such standardization in the standards organizations.

Frame format

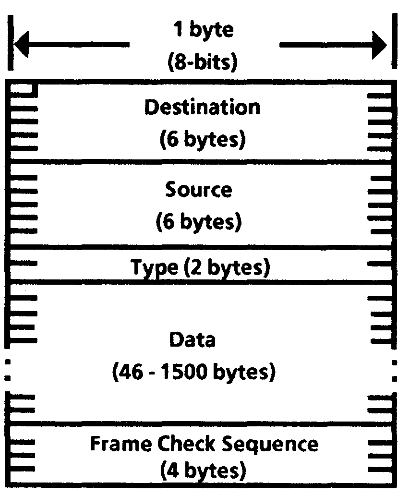


Figure 3-2 **Ethernet frame**

Figure 3-2 shows the five fields in an Ethernet data link layer frame; the source and destination address, the type field (called length in IEEE 802.3 standard), a data field containing the transmitted data, and the frame check sequence field, containing a cyclic redundancy check (CRC) value to detect transmission errors. The total length of the Ethernet frames can be 64 to 1518 bytes long. The Ethernet source and destination addresses are 48 bits long, thus uniquely identifying over 281 trillion network devices! Since a typical single Ethernet is limited to 1024 devices, why assign 48 bits when only 16 would suffice? The answer is that the Ethernet is only one component in a large internetwork system. The use of absolute (rather than relative) host numbers in an internetwork provides for reliable and manageable operation as the system grows, as machines move, and as the overall topology changes, if a local network can directly support these large host numbers. The 48-bit host number space is large enough to ensure uniqueness and provides adequate room for growth at little extra cost.

Data link layer

The data link layer itself is divided into two sublayers. The data encapsulation sublayer performs the framing, addressing, and error detection functions. The link management layer performs the channel allocation and contention resolution functions.

CSMA/CD

The general Ethernet approach uses a shared communications channel managed with a distributed control policy known as *carrier sense multiple access with collision detection*, or CSMA/CD. With this approach, there is no central controller managing access to the channel, and there is no preallocation of time slots or frequency bands. A network device wishing to transmit is said to "contend" for use of the common shared communications channel (sometimes called the Ether) until it "acquires" the channel; once the channel is acquired the device uses it to transmit a packet.

To acquire the channel, devices check whether the network is busy (that is, use carrier sense) and defer transmission of their packet until the Ether is quiet (no other transmissions occurring). When quiet is detected, the deferring device immediately begins to transmit. During transmission, the transmitting device listens for a collision (other transmitters attempting to use the channel simultaneously). In a correctly functioning system, collisions occur only within a short time interval following the start of transmission, since after this interval all devices will detect carrier and defer transmission.

To minimize repeated collisions, each device involved in a collision tries to retransmit at a different time by scheduling the retransmission to take place after a random delay period. In order to achieve channel stability under overload conditions, a controlled retransmission strategy is used whereby the mean of the random retransmission delay is increased as a function of the channel load.

Devices accept packets addressed to them and discard any that are found to be in error. It is impossible, however, to guarantee that all packets transmitted will be delivered successfully. For example, if a receiver is not enabled, an error-free packet addressed to it will not be delivered; higher levels of protocol must detect these situations and retransmit.

Taken together, the physical and data link portions of Ethernet provide the foundation for a comprehensive, sophisticated local area network capability useful in a wide variety of applications and environments.

Although Ethernet plays a central role in XNS, the architecture itself provides for a variety of ways to communicate, some of which are described below. One of the great advantages of a network architecture such as XNS is that it can be adapted to whatever communication facilities are appropriate to the application, usually without requiring major reformatting, alternative high-level protocols, and inefficient procedures.

Synchronous Point-to-Point Protocol

In addition to incorporating a flexible local area networking capability, it is also necessary for a general-purpose information system to be able to communicate over comparatively long distances.

Communication beyond local area

Fig. 3-3 illustrates this point. Consider a user who has several facilities located significant distances from each other: across a city, across a continent, or around the world. Each facility has an Ethernet local area network for communication within the facility, but the actual work done in each facility requires exchanges with the network resources and people in all other facilities. For such purposes conventional data communication techniques must be used: modems, leased or switched telephone lines, and a data link protocol specifically intended for this type of transmission.

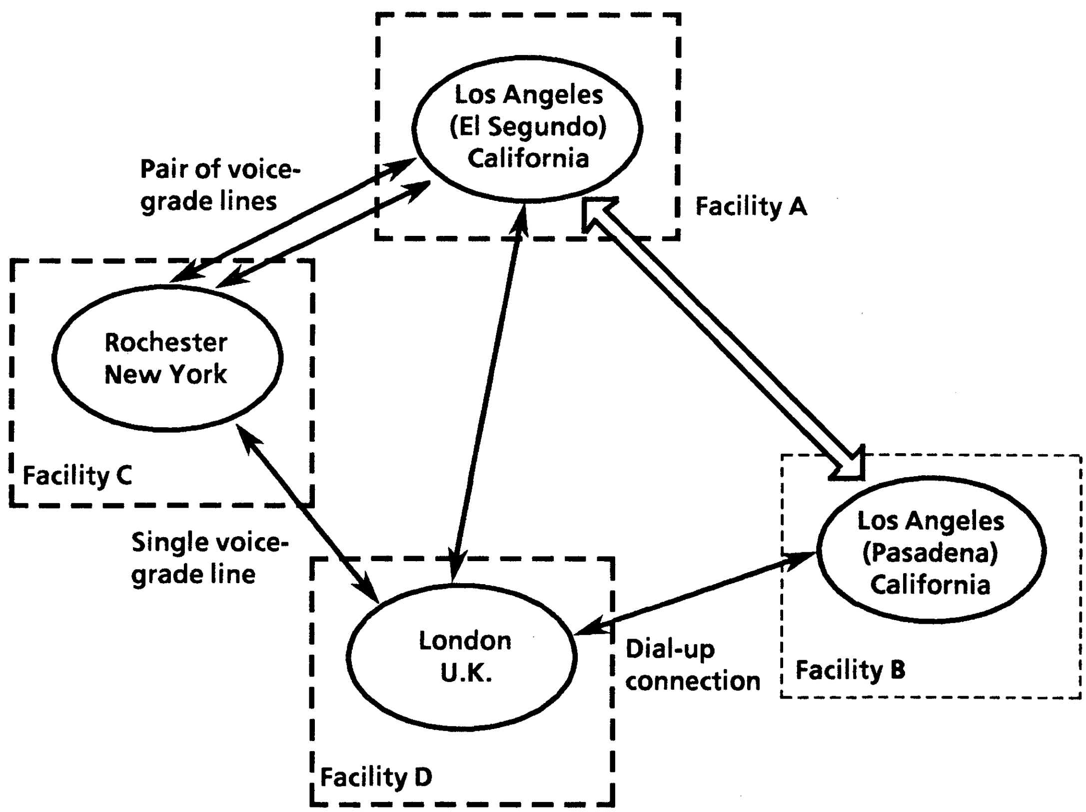


Figure 3-3 **Interfacility communication using the Synchronous Point-to-Point Protocol**

Fig. 3-3 shows a number of connection possibilities including: a single voice-grade leased line capable of supporting transmission at up to 9600 bps; a pair of such lines, where the data exchange requirement slightly exceeds the capacity of a single line; a switched (dial-up) connection, where only occasional exchanges are required or where leased lines are unavailable; and a wideband leased line, capable of operation at 56K or 230.4K bps, for comparatively high-volume exchanges.

In each case the use of synchronous modems at the end of each communication circuit is implied (when common-carrier digital services are used, special service unit devices are substituted for conventional modems). The modems or service units are interfaced with specific devices connected to each facility's local area network. The interface between these devices and the modem/service units corresponds to the interface between the data link and physical layers in the ISO Model. It is usually implemented according to the familiar provisions of EIA specifications RS-232-C or RS-449, or CCITT Recommendations such as V.4 or V.35.

Bit-synchronous transmisssion

The protocol used in XNS between remote local area networks is the Xerox Synchronous Point-to-Point Protocol, which corresponds to the data link layer of the ISO Model. Its name derives from the fact that it expects to use synchronous (rather than the lower speed asynchronous) transmission, and from the fact that it is intended for operation between two specific points rather than among a number of points (which is usually referred to as multipoint, or multidrop operation).

The Synchronous Point-to-Point Protocol is one of a class of data communication link control protocols that have come to be called bit-oriented. Essentially, this means they are capable of efficiently transmitting digital information without regard to its content, much as the public postal system functions without regard to the content of the billions of envelopes within the system. In this respect, the XNS Synchronous Point-to-Point Protocol resembles IBM's Synchronous Data Link Control (SDLC) protocol, the High-Level Data Link Control (HDLC) international standard, and others.

This XNS protocol is designed to support operations over the variety of transmission schemes illustrated in Fig. 3-3. Among its special features are:

Simplicity

The protocol defines a simple connection between communicating entities, and is easily implemented.

Transparency

In keeping with the concept of a layered architecture, the protocol makes no assumption about data that is transmitted, nor does it constrain the data in any way.

Compatibility and maintainability

A "version negotiation" provision which makes it possible for two communicating stations to agree on which version of the protocol is to be used. This is a particularly useful system maintenance feature.

Error detection

Since telephone circuits are "noisy," the protocol provides an error detection mechanism analogous to that provided with Ethernet (as in the case of Ethernet, however, error *correction* is a responsibility of a higher-level XNS protocol).

Operational flexibility

Both half- and full-duplex operation are supported over both leased and switched circuits; in full-duplex (simultaneous two-way) operation, a balanced style of transmission is provided in which neither station must be relegated to a "master" or "slave" role.

The Synchronous Point-to-Point Protocol is designed to transmit a series of frames, or packets. The organization of these packets is generally consistent with the HDLC international standard. It provides for:

—transmitting the unique XNS-standard 48-bit host address

—encapsulating and decapsulating packets up to maximum of 65,535 bytes in length (although in practice the packet length for most systems would be far less)

—several different formats for control packets, data packets, and private packets

—special sequences marking the beginning and end of a packet

—a frame check sequence, for error detection.

Relationship to HDLC

Like the Ethernet, the Synchronous Point-to-Point Protocol strives for peak performance from a normally errorless medium. When transmission errors do occur, the higher level XNS protocols provide error correction to achieve reliability. In contrast, if full HDLC were used as the point-to-point data link protocol, both HDLC and the higher-level XNS protocols would be incurring the overhead necessary to ensure reliability. The Synchronous Point-to-Point Protocol adopts the framing and error detection scheme of HDLC, but not the error correction. However, within XNS, full HDLC is also supported, as are other external protocols.

Protocol application

As a Xerox standard, the Synchronous Point-to-Point Protocol is applicable to Xerox forwarding system elements (such as internetwork routers), terminal system elements (or remote system elements), cluster system elements, and interfacing elements, as shown in Fig. 3-4. The protocol also defines a two-way alternation mode to manage half-duplex medium contention, so it is applicable to half-duplex as well as full-duplex circuits.

The Internet

To serve large and geographically dispersed office environments, Ethernet's basic communication capability must be augmented in various ways. Interconnecting multiple Ethernets will circumvent the maximum end-to-end cable length restriction, but requires mechanisms for internetwork communication. An internetwork, or internet as it is often

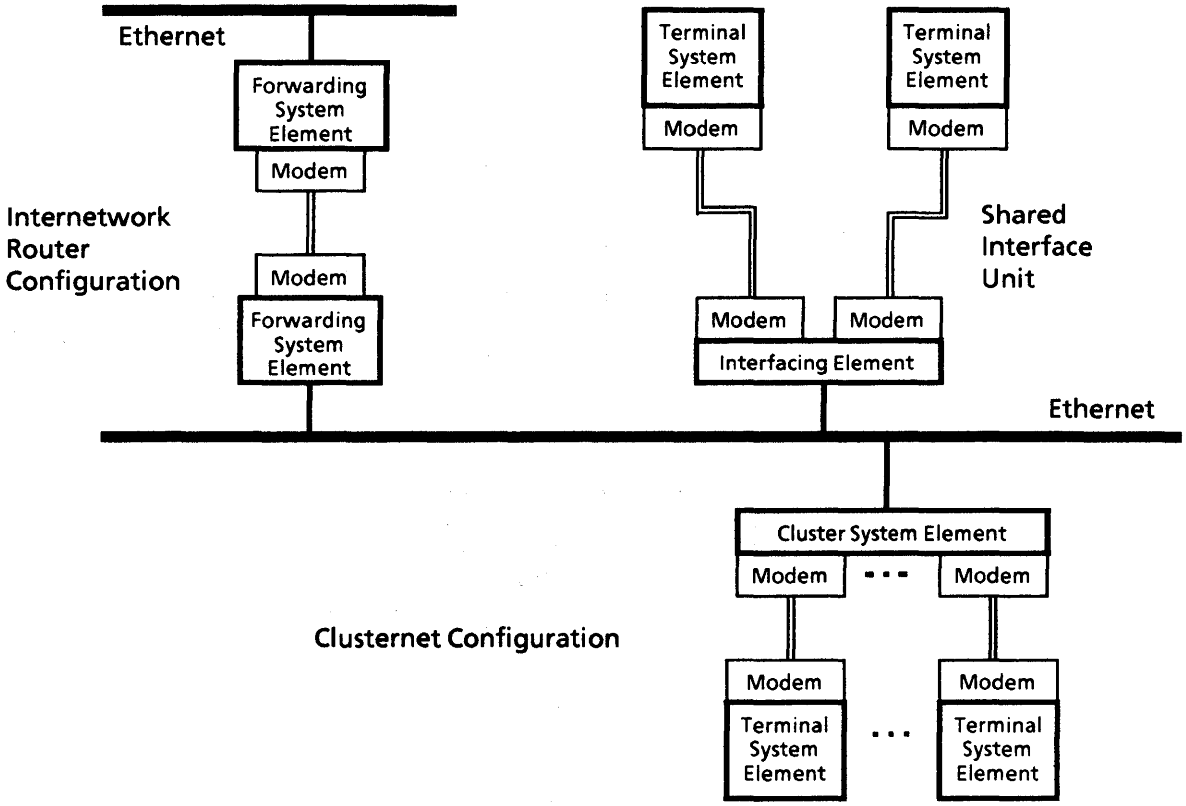


Figure 3-4 **Examples of internetworking configurations**

called, is simply an interconnection of networks. The internet architecture, consisting of the Internet Transport Protocols, provides the network and transport layer functions of the ISO Model.

Internet architecture

The Xerox internet architecture offers a richer addressing scheme and a more sophisticated routing algorithm than available in local area networks. It enables Ethernets to be interconnected by telephone lines, public data networks, or other long-distance transmission media. It also allows the communication system to be reconfigured to satisfy the immediate and future requirements of the user. For example, the Network System may have only one Ethernet initially and then be expanded (without software modification) to contain two or more Ethernets, which are interconnected directly or via other communication media.

Internet Transport Protocols

The internet layer consists of a family of Internet Transport Protocols shown in Fig. 3-5. These protocols cause information to move between sources and destinations of information in an organized and reliable manner. The several protocols are:

Internet Datagram Protocol

This defines the fundamental unit of information flow within the internet—the internet datagram packet.

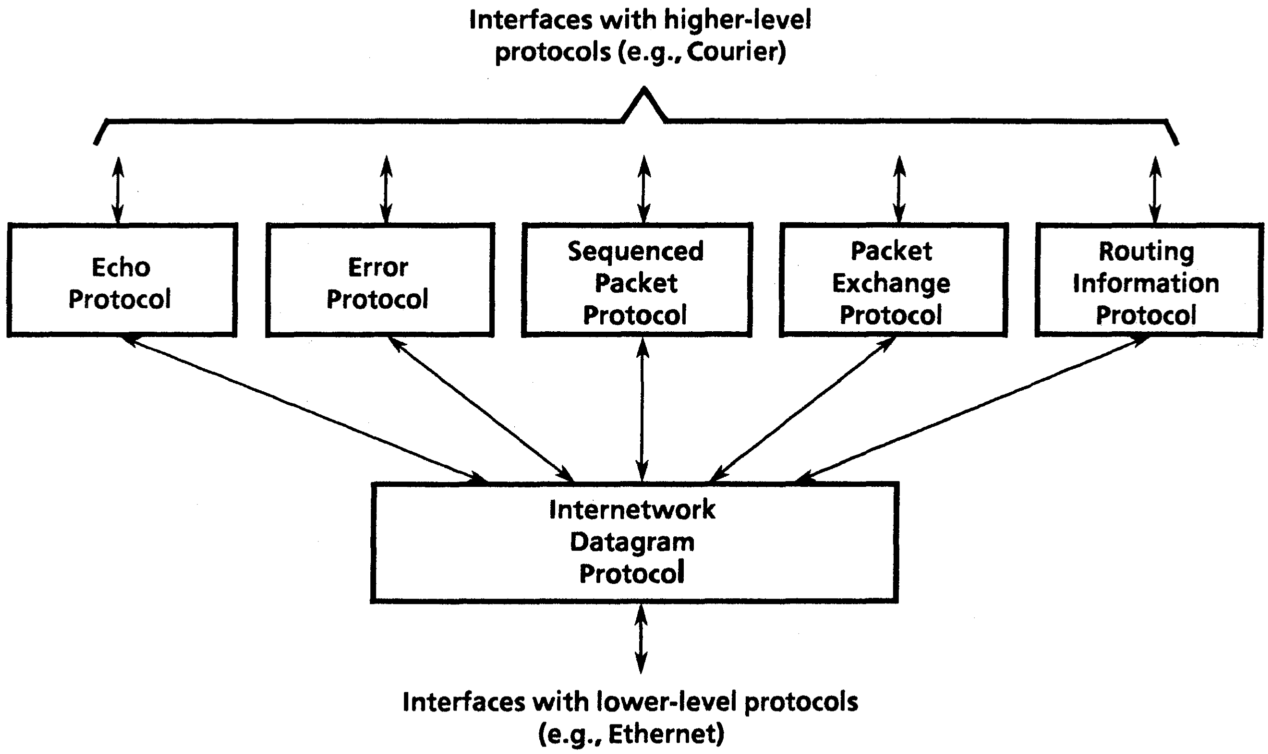


Figure 3-5 **The XNS Internet layer**

Sequenced Packet Protocol This provides for reliable, sequenced, and duplicate-suppressed transmission of a stream of packets.

Packet Exchange Protocol This supports simple transaction-oriented communication involving the exchange of a request and its response.

Routing Information Protocol This provides for the exchange and dissemination of inter-network topological information necessary for the proper routing of datagrams.

Error Protocol This standardizes the manner in which low-level communication or transport errors are reported.

Echo Protocol This is used to verify the existence and correct operation of a host, and the path to it.

The work done by Internet involves interaction between one or more of the specialized protocols and the basic Internetwork Datagram Protocol. To simplify this discussion, functional distinctions between the operation of these various protocols have been avoided. The Internet documentation described in appendix D contains more detail about this relationship.

Datagrams

The XNS internet architecture identifies a fundamental unit of information flow called an internet packet or "datagram." It refers to a condition of "self sufficiency" for the transmission unit of information, the packet. A datagram, therefore, is a packet (typically several hundred bytes of information) whose movement through the system is individually controlled. (This is in contrast to systems that use the concept of a "virtual circuit" in which a logical relationship is constructed between source and destination for purposes of transmission; in such systems, the flow of packets is typically managed on a group basis.)

Packet structure

The Internet packet contains control information and data as shown in Fig. 3-6.

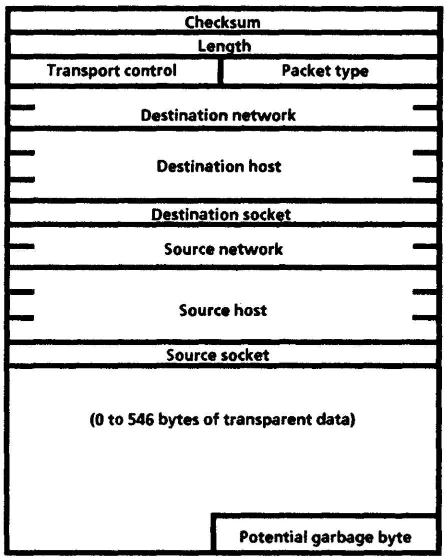


Figure 3-6 **Internet packet or "datagram"**

The Internet packet fields fall into three categories: adressing fields, which specify the address of the destination and source network addresses; control fields, which consist of checksum, length, transport control, and packet type fields; and data fields, which carry the data and consist of information that is interpreted only at the next higher Courier layer.

The function of the Internet is thus to manage the flow of these datagrams through the network. The "network" in this context consists of the local network, plus all other connected networks, however far-flung they may be. Internet is responsible for maintaining a global perspective. Regardless of how large or small the network may be, the process of managing datagram flow through it is uniform.

Source and destination addresses

The Internet packet has source and destination address fields which is more general than the 48-bit host number used on the Ethernet. It also includes a 32-bit network number and a 16-bit socket number for the source, as well as the destination address, as shown in Fig 3-6.

Host number

A host is any system element that supports the XNS communication protocols and is connected to a network. In XNS, every system element is assigned a unique 48-bit number independent of the network to which it is connected. Xerox chose an absolute host numbering scheme instead of the more conventional network-specific host numbering scheme. Absolute host numbers have many advantages when building large distributed systems. Operating systems and application software can use this number in generating unique identifiers. Also, when a host is moved from one network to another, its host number does not change, making alterations to the hardware or special bookkeeping unnecessary. Since such alterations are required when using network-specific host numbers, use of absolute host numbers substantially reduces field service overhead.

Xerox internets consist, for the most part, of Ethernets, which is the main reason that Ethernet addresses are identical to 48-bit host numbers. This structure is strictly for convenience and in no way compromises the generality of the architecture. When a host is connected to more than one Ethernet, its 48-bit Ethernet address on all Ethernets is equal to its 48-bit host number.

Xerox has implemented a procedure whereby other implementors of XNS hosts may reserve a block of host identification numbers. (Note that since the host number is 48 bits long, this represents an enormous address space: if all the numbers were used, approximately 281 trillion hosts would be identified! In practice, however, the number of hosts is limited to about four billion.)

Network number

Since a host number uniquely identifies a specific host, the network number field would seem redundant, but it is needed for internetwork routing. When the network number is included in the network address, each host has to know only the (partial) path to each network rather than that to each host—significantly reducing the amount of information that must be retained. A host may be connected to more than one network but still has a unique identity.

An internet packet addressed to a host contains the identity of the network to which the source believes the host is connected. Internetwork routers attempt to route the internet packet to the host via this network. If no route to the specified network exists, the packet is not delivered and client software must use another network address. The higher-level Clearinghouse supplies all network addresses for resources such as file servers, print servers, or a user's mailbox. All networks within an internet have unique network numbers.

Socket number

A socket is a uniquely identified data structure within a host, to which internet packets can be delivered, and from which internet packets can be transmitted. A socket is inherently a bidirectional structure, able to both send and receive packets.

The internet delivers packets (datagrams) among sockets in much the same way that the post office transfers letters between post office boxes. The sockets may be on the same host, on hosts on the same network, or on hosts on different networks. A host that receives an internet packet first delivers the packet to the approximate socket, and then the client of the socket demultiplexes the packet according to its transport protocol type. In this respect, the XNS approach differs from that used in other internetwork architectures such as the one defined by the ARPA Internet Protocol, which does not include a socket number in its network address—a host receiving an

ARPA internet packet demultiplexes it according to its protocol type, and the next higher level of protocol then has the option of defining a socket-like object.

Sockets are numbered within a particular host. The socket number is a 16-bit binary value (providing 65,536 possible socket numbers). The first 3000 socket numbers are reserved for well-known sockets; that is, the service performed by software using these sockets is well-known. Each host supplying a specific well-known service does so at the same well-known socket. All other socket numbers can be reused.

Multidestination addressing

Multicast is the delivery of a packet to more than one destination, and it can be performed at either the internetwork or intranetwork level if the transmission medium supports the concept. XNS supports internet multicast. Broadcast is a special case of multicast in which a packet is delivered to all hosts in the internet. The need for a generalized multicast capability arises from the anticipated need for a more general addressing capability. Broadcast is used in many situations to search for an object or to inform all hosts of an event. Although all applications can be designed without this capability, multicast provides some performance improvements.

Internet delivery and routing

One of the major responsibilities of the Internet layer is routing and delivery of information from a source to a destination. Since the network over which this is to be accomplished may be very large, involving many different subnetworks with a variety of interconnecting links, the routing procedure is very generalized. It presumes that intermediate functions, performed by internet routers, assist in the process. Internet routers move datagrams through the network based on the idea that each routing element in the network understands ***the next place to send the datagram***. Depending on the scope of the network, many or all of the routing provisions in the chain may not "know" the entire network; but with the "next place to send" approach, the network can be arbitrarily large and the datagram will arrive at its destination.

Encapsulation and decapsulation

The internet packet is usually encapsulated for transmission through the various communication networks; the encapsulation specifies the addressing and delivery mechanisms specific to that network. Each communication network may have a different form of internal addressing. When an internetwork packet is to be transported over a communication medium, the immediate destination of the packet must be determined and specified in the encapsulation. The immediate destination is determined directly from the network address if it is the final destination, or through a routing table if it is an intermediate destination.

Routing

Basic routing is undertaken by the Internet Datagram Protocol. Each host connected to the network is equipped to execute this protocol. It uses a routing table containing the addresses of other elements in the vicinity of the host in question, specifically including the internetwork routers responsible for directing datagram flow between subnetworks. This routing information is kept up to date by the Routing Information Protocol.

Internet packets are routed through the internet using a store-and-forward algorithm that relies on the routing table. Routing information is maintained through the use of adaptive procedures. Neighboring internetwork routers periodically exchange routing information. Changes in internetwork topology may cause the routing tables in different internetwork routers to become momentarily inconsistent, but the algorithm is stable since routing tables rapidly converge to a consistent state and remain that way until the topology changes again.

A host that is not an internetwork router obtains routing information by polling internetwork routers on its directly connected networks. The host may obtain updates periodically if it receives the broadcast packets that other internetwork routers are exchanging; if not, then it may periodically poll internetwork routers for updates. If more than one internetwork router is providing paths to other networks, an internetwork router or host can merge the information it receives and thus select the best route for packets directed to any network.

When internet packets traverse other communication networks that do not support XNS absolute host numbers, like the Bell Telephone DDD network, Telenet, or other public or private data networks, translation tables will have to exist in the necessary hosts and internet routers to perform translation from absolute host numbers to internal addresses. This does not cause many operational problems other than setting up and maintaining these translation tables as appropriate.

Message integrity

While the basic unit of information in the Internet is the datagram, or internet packet, the overall purpose of communication is to deliver entire messages from source to destination. A message consists of an arbitrary number of datagrams—whatever is required to send the complete message. In some cases this can be a fairly large number, as in the case of a file containing a scanned photographic image, which might consist of as many as 1000 500-byte datagrams (a customary datagram size).

A basic characteristic of all packet-switching communication systems (including XNS) is that the freedom to optimize the routing of the individual packets can cause a transmitted string of packets to arrive at their destination out of sequence with respect to their order at the source. This happens as individual packets are routed through different links, with different transmission delays, and as some packets are retransmitted due to error conditions. It is also possible for packets to be duplicated and, in rare cases, to vanish altogether. Left uncorrected, this situation means that packet-switching systems have a number of opportunities for the integrity of messages to be violated. This is clearly unacceptable.

Sequenced Packet Protocol

The XNS Internet layer deals with this problem through the functions of the Sequenced Packet Protocol (Fig. 3-5). A sequence number is assigned to each transmitted packet within the Internet layer at the source host. This number is checked within the Internet layer at the destination host, and if aberrations (misordered packets, duplicates, missing packets) are detected, corrective action is taken on a negotiated basis between the source and destination Internet layers.

To achieve this result, Internet provides for associating the source and destination sockets in a connection, a temporary relationship in which various details concerning the operation of the exchange are negotiated and in which sequence numbers, acknowledgments, and related supervisorial information are exchanged.

This connection between source and destination can be thought of as establishing a virtual circuit between them, roughly analogous to what takes place in X.25 wide-area switched packet networks. The Sequenced Packet Protocol should be seen as using a datagram-based lower-level protocol, the Internetwork Datagram Protocol, to create a virtual circuit between source and destination, by taking responsibility for the integrity of information flow.

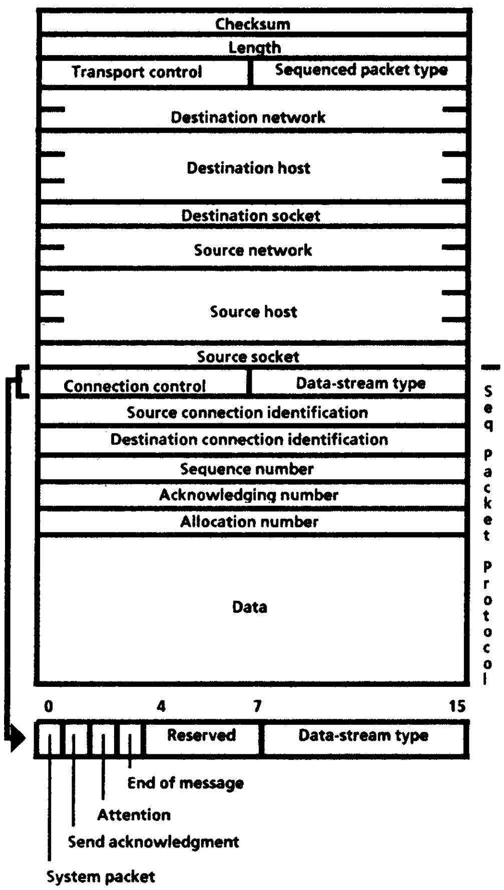


Figure 3-7 **A sequenced packet protocol packet allows successive transmission of internet packets.**

Arranging packets into messages and message sequences also circumvents the packet-size limitation at lower levels of the protocol architecture. The Sequenced Packet Protocol provides a mechanism to punctuate the stream of packets with end-of-message boundaries. The protocol specifies the format of packets as shown in Fig. 3-7, and the meaning of packet sequences.

The connection control field contains four bits that control the protocol's actions: system packet, send acknowledgment, attention, and end-of-message. The system packet bit enables the recipient to determine whether the data field contains client data, or is empty and the packet has been sent only to communicate control information required for the connection to function properly. If the send acknowledgment bit is set, the source wants the receiver to acknowledge previously received packets.

In a distributed environment, special procedures must be provided to bypass the normal flow control and interrupt a process. If the attention bit is set, the source client process wants the destination client process to be notified that this has arrived. If the end-of-message bit is set, then the packet and its contents will terminate a message and the next packet will begin the following message.

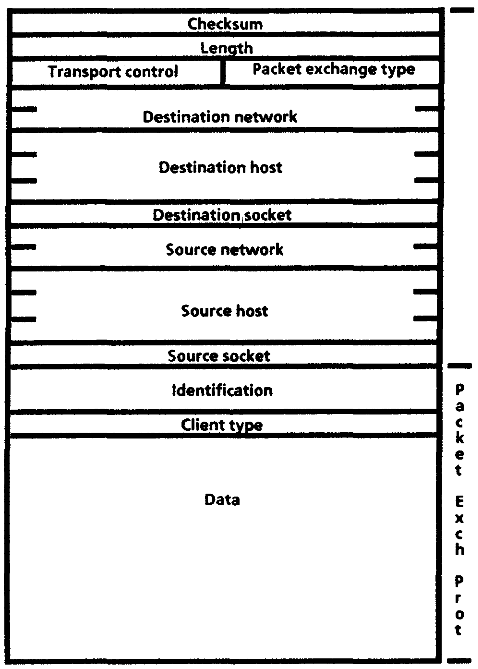


Figure 3-8 A packet exchange protocol packet simply transmits a request and receives a response.

The primary bridge between this protocol and the next layer is the data stream type field which provides information that may be useful to higher-level software in interpreting data transmitted over the connection. A connection must be created before it can be used and discarded when no longer required. One end of a connection is said to be established when it knows the address (host and socket number) and connection identification of both ends of the connection.

Connection-oriented communications, which is supported by the Sequenced Packet Protocol, involves an extended conversation by two machines in which much more information is conveyed than can be sent in one packet going in one direction. For simple transaction-oriented communication which involves single requests and responses, a simpler Packet Exchange Protocol is also provided in the internet architecture.

Transmitting a request in a packet and receiving a response via the Packet Exchange Protocol (Fig. 3-8) will be more reliable than transmitting internet packets directly as datagrams, but less reliable than the Sequenced Packet Protocol.

There are only three fields in the packet. An identification field, which contains a transaction identifier, is the means by which a request and its response are associated. A client type field indicates how the data field should be interpreted at higher levels. A data field contains whatever the higher-level protocols specify. Such a protocol might be used in locating a file server through a resource-location service, such as the Xerox Clearinghouse.

Other Internet protocols

As dominant as the Sequenced Packet and Packet Exchange Protocols are in the internet layer, they do not handle everything. Other internet protocols are required for specialized tasks of routing, error reporting, and diagnostics.

Routing Information Protocol

The Routing Information Protocol provides for the exchange of topological information among internetwork routers and workstations. Two packets are defined by the protocol: one of them requests routing information, and the other supplies it. The information supplied is a set of network numbers and an indication of how far away those networks are. This information is either sent on specific request or periodically distributed by all internetwork routers, which use the data to maintain routing tables that describe all or part of the internetwork topology.

Error Protocol

The Error Protocol is intended to standardize the manner in which low-level communication or transport errors are reported. Moreover, it can be used as a debugging tool. If, for example, a machine receives a packet that it detects as invalid, it may return a portion of that packet by means of the Error Protocol, along with an indication of what is wrong. If the packet is too large to be forwarded through some intermediate network, the error protocol can be used to report that fact and to indicate the length of the longest packet that can be accommodated. If too many of these return, the system designer may conclude that something is wrong with his implementation.

Echo Protocol

Another useful diagnostic and debugging tool is the Echo Protocol, which is used to verify the existence and correct operation of a host, and the path to it. It specifies that all Echo Protocol packets received shall be returned to the source.

Internetwork Routing Service

The Internetwork Routing Service (IRS) links geographically dispersed networks into a single internet by connecting a local network to other networks via telephone lines or public data networks, It thus enables communication with remotely-located XNS system elements.

The IRS implements the Internet Transport Protocols, and its functions include packet forwarding, routing decisions, and the gathering of routing information. Of particular importance is the media flexibility that allows XNS-supported systems to make use of the best suited transmission facilities: dedicated circuits, switched circuits, and public data networks. In addition, the Internetwork Routing Service makes it possible for isolated remote devices (i.e., those established on a standalone basis without their own Ethernet local area network, servers, etc.), or clusters of such devices, to interface with local networks via telecommunication circuits.

The Internetwork Routing Service implements one of the most important characteristics of Xerox Network Systems: geographic independence. This means that users of XNS in one location can communicate with users and network resources in a remote location, *without being aware of, or having to take special measures because of the distances between the two locations*. In other words, an XNS network appears as a single entity to a user, not a series of interconnected local area network segments.

How IRS works

The IRS is fundamental to all installations with multiple interconnected Ethernets. When a local network device sends information to another network, it sends the information to an IRS which forwards it to its destination (see Fig. 3-9).

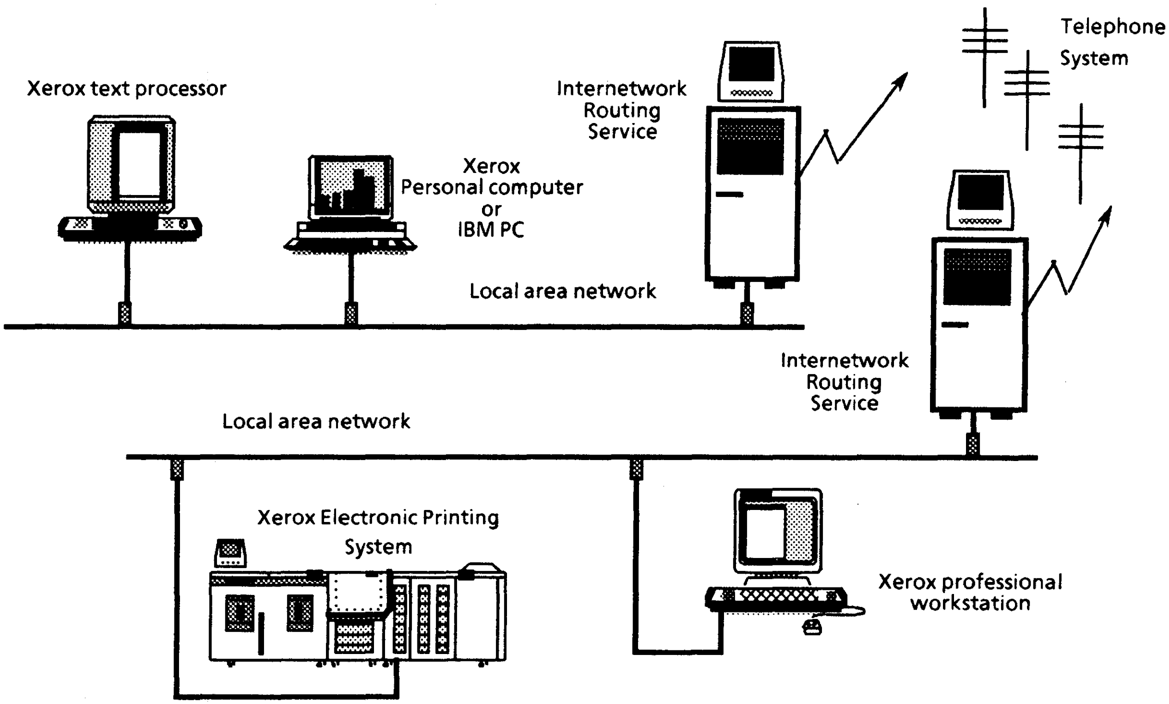


Figure 3-9 **Two local area networks interconnected by a communication line**

To accomplish the forwarding function, each IRS holds a complete map (routing information tables) of the internet. This map contains the address of each remote network, how far away it is, and the next IRS along the path to get there. When an IRS receives a packet to be forwarded to a remote network, it uses its map to find the address of the network and sends the packet along its way using the shortest path to that network. IRS's on an internet exchange their maps on a regular basis. After an IRS is activated, it gradually learns of the complete internet map from neighboring IRS's by means of this exchange. Likewise, changes to the internet gradually propagate from one IRS to another until all IRS maps reflect the change. No manual intervention is needed.

The operation of the Internetwork Routing Service is invisible to the user of a local workstation. On the Xerox professional workstations, users access remote services through icons on the desktop in exactly the same way they access local services. For example, to access a file drawer on another network, a user would open the Directory icon and bring a copy of the file drawer icon to his desktop. The remote file drawer could then be accessed in the same way as a local file drawer. Although the user commands on other workstations may be different, the functioning of the IRS is still invisible. The user of any workstation never needs to communicate directly with the IRS.

Dedicated and switched circuits

Dedicated telecommunication circuits are generally available on a full-duplex basis at speeds to 56 kbps. The interface to these circuits is by means of the RS-232-C standard, or its higher-speed equivalents; use of these circuits requires corresponding modems. The Synchronous Point-to-Point Protocol is designed to operate over a dedicated circuit.

Switched circuits consist of telephone dial-up and similar facilities. In essence, a switched circuit provides a temporary point-to-point connection. The Internetwork Routing Service is capable of operating with switched connections made manually or of automatically dialing a connection. In both cases, subsequent operation over the circuit is similar to operation over a dedicated circuit.

X.25 public data networks

An alternative to the use of conventional telecommunication circuits is the use of public data networks (usually packet-switched), which provide user interfaces that comply with CCITT's Recommendation X.25. In some parts of the world, Europe in particular, X.25 public data networks are much more widely available than dedicated circuits and are able to overcome many of the network interconnection and trans-border data flow problems. The Internetwork Routing Service makes it possible for individual Ethernet systems to be linked by means of these public data networks. In such arrangements the X.25 Virtual Circuit Protocol, rather than the Synchronous Point-to-Point Protocol, is used at the link control layer.

An X.25 Internetwork Routing Service is a powerful facility. Means are provided for a local or remote system administrator to initiate and terminate individual X.25 links; to interrogate the status of a given link; to maintain a log of calls; and to automatically retry failed outgoing calls (as happens when a public data network's local ports are momentarily tied up). The Internetwork Routing Service also provides basic security checks on incoming calls to ensure that only authorized calls are accepted.

When information is transmitted using X.25 circuits, the IRS wraps the data in a special X.25 protocol and sends it to the local X.25 network. The network consults its routing table and sends the information to the destination IRS. The X.25 network serves as a pass-through medium for directing information. When the IRS receives the information, it unwraps the data and routes it to its final destination.

The use of X.25 networks does not preclude the use of other communication media as shown in Fig. 3-10. X.25 networks simply provide another, often more desirable, alternative for internet communication.

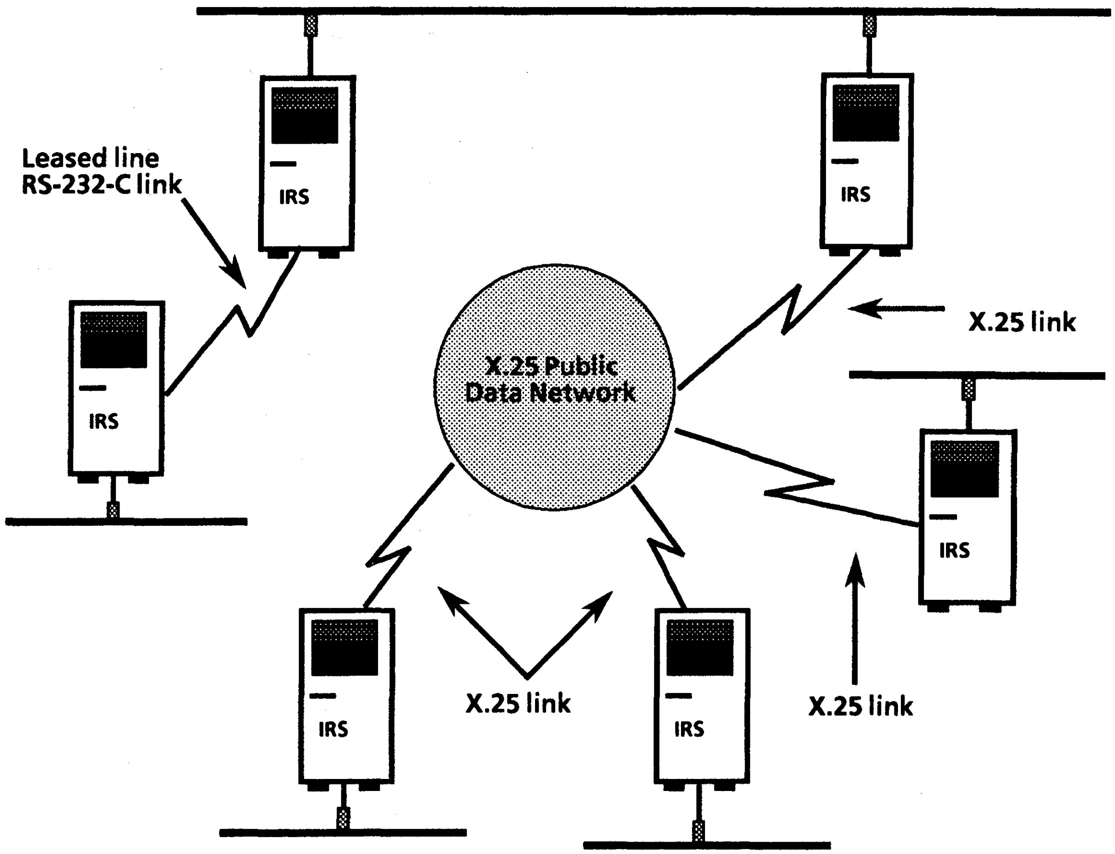


Figure 3-10 **Internet Routing Service (IRS) with X.25 links**

Clusternet communication

The IRS also makes it possible for individual workstations or a cluster of workstations (equipped with appropriate communication interfaces) to communicate with each other and the internet without the use of separate communication servers (see Fig. 3-11). The Synchronous Point-to-Point Protocol is used between Internetwork Routing Service and the workstations, and the resulting network is called clusternet.

The IRS includes a clusternet router that provides routing information to the clusternet. This router uses the network number of the clusternet and the host numbers of the remote workstations to route information to and from the clusternet's ports.

The remote workstation is attached to the IRS using an RS-232-C connection via leased lines or dial-up capability. In dial-up operations, the remote workstation becomes networked only when the user dials an IRS clusternet port. For the remainder of the dial-up session, the user can access all of the services normally provided by a direct connection to the network.

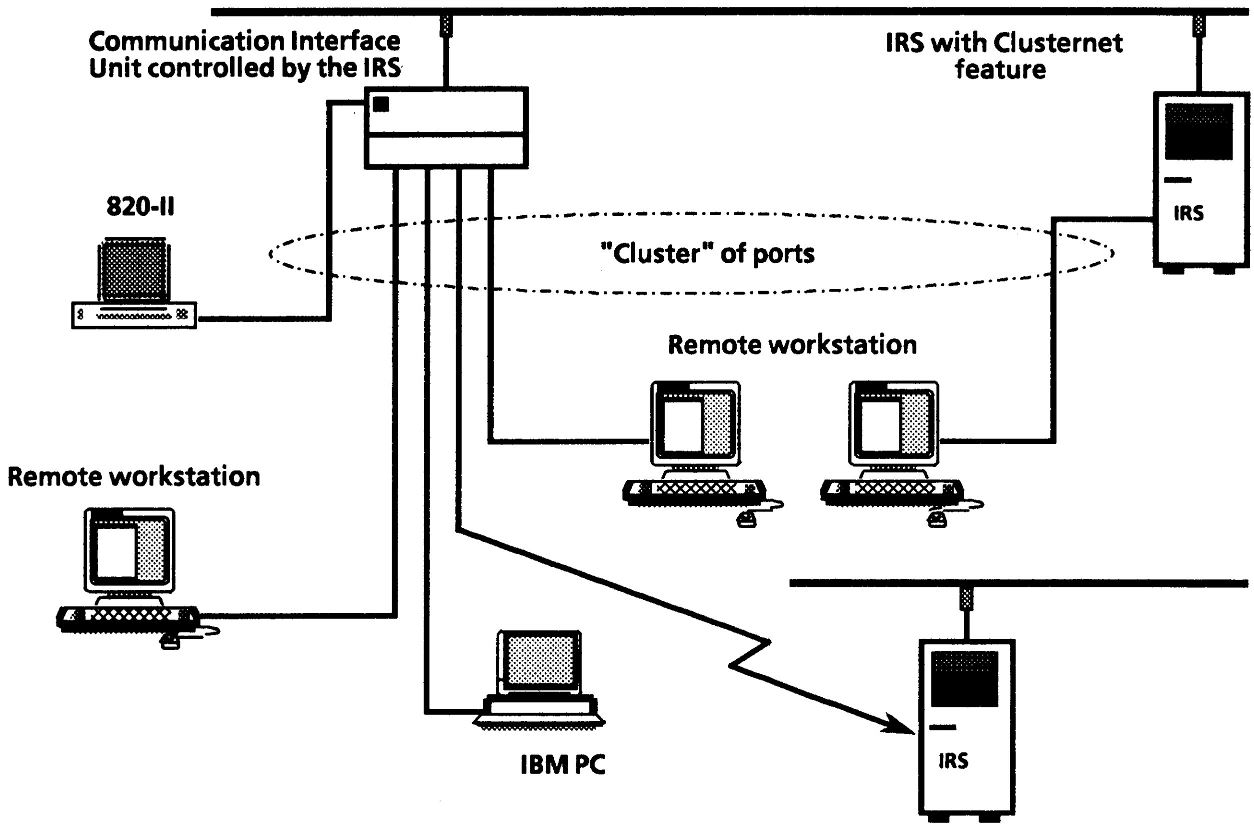


Figure 3-11 **Internet Routing Service (IRS) with clusternet feature**

Network management

Network management has traditionally been associated with managing telephone systems and modems. A network manager might be responsible for determining who gets what equipment and with what access rights, for publishing an internal telephone directory, diagnosing problems, and for reconfiguring the system to meet customer needs. The internet, including all its diverse workstations and services, must be similarly managed. However, the problem of managing an internet is more a problem of managing a distributed system. Managing the communication components is a subset of the problem.

Service monitoring

Network management in the distributed system involves monitoring the services and communication components of the internet. While most networks support monitoring only from a central network control center, entities in XNS can be monitored from any system element in an internet. Controlled access availability to the network management tools prevents their unauthorized use. The monitoring of services includes performance and load statistics, and unusual events. Monitoring communication components include statistics that assist in problem determination, plus network planning and configuration.

The Server Monitor Service (SMS) is a program that watches over servers on the internet, assesses their availability, and reports problems through mail messages. SMS runs on an XNS server and interacts with a specified set of servers. It maintains a database of information which includes the configuration of the target servers being monitored, the frequency with which a given server should be polled, and lists of interested users. These lists are used to send messages to interested users whenever a server's availability changes. All the information obtained about a server's condition is recorded by the SMS and can be examined.

System administrators

A large internet configuration (which may have thousands of users and their associated workstations, printers, file servers, mail servers, etc.) requires considerable management. To facilitate this, a system administrator function is provided in XNS that allows designated users to be registered as system administrators.

The XNS distributed system approach allows system administrators to provide localized system installation and reconfiguration. Alternately, the system administrator function may be performed from a remote network or a remote workstation. Network integrity is maintained by the strong emphasis on security in the XNS architecture. This gives great flexibility in network management and control.

4. REMOTE PROCEDURES: COURIER

Xerox Network Systems uses a model in which a system exists to perform useful work. The purpose of the communication infrastructure is to facilitate doing that work. Although other forms of information exchange are also used, a key part of the XNS architecture deals with how the exchanges take place between a service provider and a service consumer that cause work to be done at a location. This location may be at a distance of thousands of miles or within the same processor. To the architecture, there is no significant difference.

It is crucial to the implementation of a modern distributed system that these work exchanges be accomplished efficiently, with maximum flexibility, and with no loss of generality.

This exchange is at a middle layer in the architecture in between the application-layer protocols and the network/ transport layer protocols. The functions involved at this point are more primitive than application functions (but still vital).

A special protocol called Courier (subtitled The Remote Procedure Call Protocol) defines the method by which directions for accomplishing work within XNS are sent and appropriate responses returned. The functions performed within Courier relate approximately to those in the session and presentation layers of the ISO Model.

The Courier model

Courier facilitates the construction of distributed systems by defining a single request-reply or transaction discipline for an open-ended set of higher-level application protocols such as printing, filing, and mail. Courier specifies the manner in which a workstation or other active system element invokes operations provided by a server or other passive system element (see Fig. 4-1).

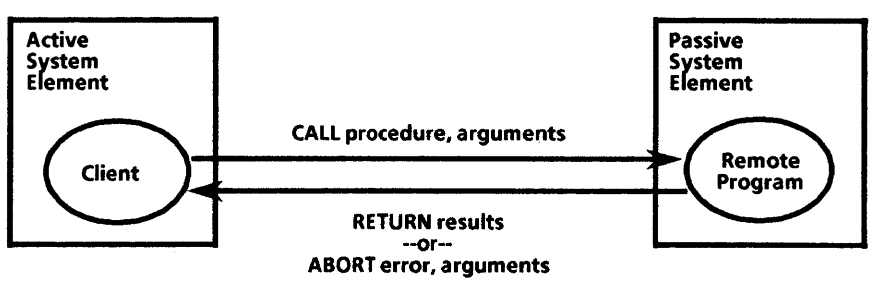


Figure 4-1 **The Courier remote procedure call model**

Courier is based on the concept that an active system element issues call procedures which contain "arguments" (data items or input specific to the requirements of the called procedure) necessary to get the work done. The remote procedure is undertaken in a passive system element and the result of that work is returned to the active element. If something goes wrong, the procedure is aborted and an error statement is returned. The error statement contains "arguments" (data feedback) that will help the calling element determine what went wrong.

Courier as a language

Courier does for distributed system builders some of what a high-level programming language does for implementors of more conventional, non-distributed systems. Pascal, for example, allows the system builder to think in terms of procedure calls rather than in terms of base registers, save areas, and branch-and-link instructions. So Courier allows the distributed system builders to think in terms of remote procedure calls, rather than in terms of socket numbers, network connections and message transmission. Courier also provides a rich set of predefined as well as constructed data types including boolean, integer, cardinal, string, array, and record among others.

Not all transaction-oriented communication may be best accomplished using Courier. Some applications may necessitate the use of datagrams rather than virtual circuits (upon which Courier is based). The XNS protocols at the Internet layer support applications for which Courier is inappropriate.

The internal Courier layers

Courier is internally divided into three hierarchical layers (see Fig. 4-2): The block stream at the lowest layer, the object stream at the middle layer, and the message stream at the highest layer. The block stream layer carries blocks of arbitrary binary data between system elements in accordance with the Internet Sequenced Packet Protocol. The object stream layer carries structured data (such as booleans and cardinals) between system elements. The message stream layer carries service requests (call messages) and replies (return and abort messages) between system elements.

Courier assumes that a higher-level function (such as an XNS application layer or a user process designed to interface with XNS at the Courier level) issues the appropriate remote procedure calls in the form of message streams. Courier's responsibility is to interface with the Sequenced Packet Protocol within Internet to move the request to the appropriate destination. Courier can thus be seen to have two primary functions: translating specific remote procedure calls into a common "language" (in the computer sense of the word) for subsequent action, and directing the communication system, as represented by Internet, to convey the required instructions to getting the work done. A third function of Courier is bulk data transfer, to which it has been adapted (as discussed in the next section).

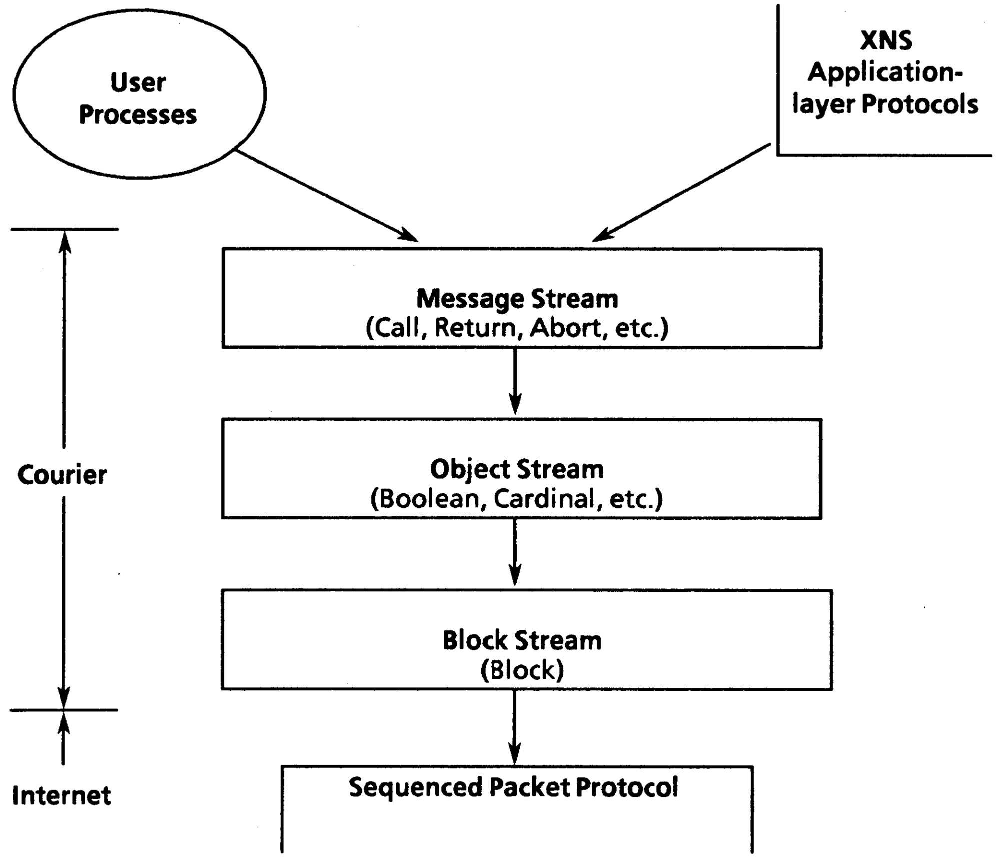


Figure 4-2 **The layers within Courier**

Because the top layer within Courier accepts application-generated messages and translates their content into a more primitive, but useful, form, the relationship between the *input* to Courier and the *content* of Courier may be considered similar to the relationship between a source and object computer language. In general, a source language is responsible for expressing the purpose of the program in terms particularly compatible with the purpose of the originator; the object language expresses that purpose in terms particularly compatible with the machine responsible for accomplishing that purpose. In this case, the "machine" is the Sequenced Packet Protocol and its underlying Internet and transmission protocols. Note that this source/object perspective is consistent with the functionality at the ISO Model's presentation layer which is encompassed by Courier in the XNS structure.

For these reasons Courier is considered a language as well as a protocol. The transaction-oriented expressions which Courier accepts are often spoken of as "written in Courier." The notation used for the Courier language is in Backus-Naur Form

(BNF). This is a formalized methodology for writing computer language statements produced in conjunction with the development of the ALGOL 60 language in the 1960s.

To see how Courier is utilized, consider a user who wishes to retrieve a file from a file server on the Internet. Let us assume that the user is on a non-Xerox workstation and on a remote network thousands of miles away. The file system on the server contains named directions, each of which comprises one or more files. For this operation, the user would need a file transfer protocol. Utilizing Courier's standard notation, it is very easy to formally specify the hypothetical file transfer protocol shown in Fig. 4-3. Remote procedures are provided in this example for gaining and relinquishing access to directories and for storing and retrieving files.

```
SimpleFile Transfer: PROGRAM 13 VERSION 1 =
BEGIN
- types
Credentials: TYPE = RECORD (user, password: STRING);
Handle: TYPE = UNSPECIFIED;

- procedures
OpenDirectory: PROCEDURE (name: STRING, credentials:
   Credentials)
   RETURNS (directory: Handle) REPORTS (NoSuchUser,
   IncorrectPassword, NoSuchDirectory, AccessDenied) = 1;

   Store File: PROCEDURE (name: STRING, directory: Handle)
   REPORTS (NoSuchFile, InvalidHandle) = 2;

   RetrieveFile: PROCEDURE (name: STRING, directory:
      Handle)
   REPORTS (NoSuchFile, InvalidHandle) = 3;

   CloseDirectory: PROCEDURE (directory: Handle) REPORTS
   (InvalidHandle) = 4;

- errors
NoSuchUser:                ERROR = 1;
NoSuchDirectory:           ERROR = 2;
NoSuchFile:                ERROR = 3;
IncorrectPassword:         ERROR = 4;
AccessDenied:              ERROR = 5;
InvalidHandle:             ERROR = 6;
END.
```

Figure 4-3 **Example of Courier usage**

To retrieve the file, the user's workstation locates and then establishes a connection to the file server. The workstation opens the directory, retrieves the file, and closes the directory. The workstation then terminates the connection. The workstation opens and closes the directory by calling the remote procedures named OpenDirectory and CloseDirectory in the file server. It requests retrieval of the file by calling the remote procedure named RetrieveFile, which tells the file server of the intention to retrieve. As soon as that procedure returns, the file server transmits the contents of the file on the connection, using the bulk data transfer mechanism.

Bulk data transfer and third parties

Courier supports applications whose communication requirements are primarily request/reply transactions. Of course, not all communication exchanges are transaction-oriented. An important alternative category involves the movement of comparatively large quantities of data (e.g., an entire document file) which would be inefficient to send as an argument within a procedure call. To allow for this, XNS provides a special adaptation of Courier called the Bulk Data Transfer Protocol which provides a single bulk data transfer discipline for an open-ended set of higher-level application protocols.

Bulk data is an arbitrarily long sequence of eight-bit bytes optionally interpreted as a single Courier data object. Although any data may be modeled as bulk data, bulk data transfer is intended primarily for transporting objects that may be too large to be reasonably modeled as arguments or results of remote procedures. For example, directory listings and the contents of files are among the entities that might be appropriately modeled as bulk data.

The Bulk Data Transfer Protocol is itself a Courier-based protocol, and it standardizes the manner in which the sender and receiver of bulk data make contact with one another; how bulk data is demarcated; and how the transfer can be aborted, if necessary, by either party.

The transfer of bulk data conceptually involves an initiator which requests the transfer, a sender which *produces* the data and a receiver which *consumes* the data. The initiator is often the sender or receiver. The Bulk Data Transfer Protocol supports three forms of bulk data transfer: Third party, immediate, and null. A *third-party* transfer is required for the most general case, in which the initiator is neither the sender nor the receiver. An *immediate* transfer suffices in the most common case, in which the initiator is either the sender or the receiver. A *null* transfer handles the degenerate case in which the data transfer that would normally take place must be suppressed.

Third Party transfers

Unlike simple remote procedure calls, bulk data transfers may involve third parties. For example, a workstation might wish to cause a file server to send a file to a print server for printing. Fig. 4-4 illustrates this third-party exchange and provides a general idea of its operation. In this example, a third party initiator wishes to cause information to flow from a sender to a receiver. It does this by issuing special procedure calls called *Produce* and *Consume* (Step 1, A and B). Assuming for a moment that the receiver is the active controlling party, it issues a Bulk Data Send procedure call to the passive (controlled) sender (Step 2). At that point, the sender actually transmits the bulk data (Step 3), and issues a return message to the receiver (Step 4). Finally, the sender and receiver issue returns to the initiator (Step 5, A and B). The Bulk Data Transfer Protocol permits either the sender or receiver to be active elements, depending on their characteristics and the nature of the data being transferred.

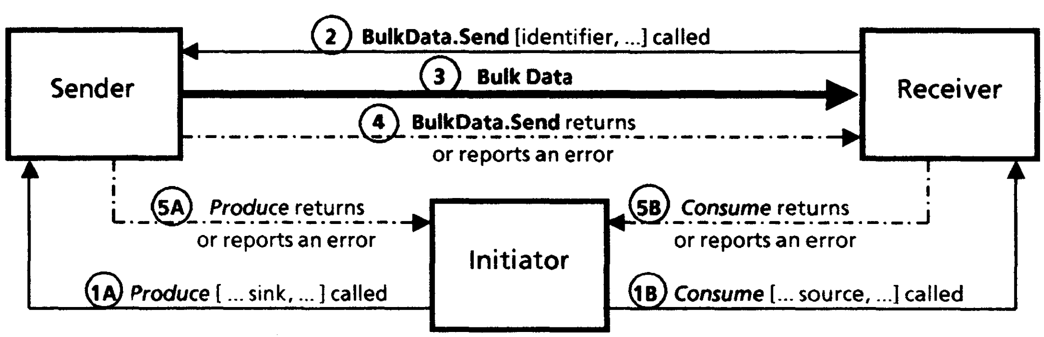


Figure 4-4 **Third-party bulk data transfer (receiver active)**

Immediate transfers

A third party is not always involved, of course. The initiator is often the sender or receiver, and this becomes a much simpler case. The initiator effects an immediate transfer by calling *Produce* or *Consume*, whichever is appropriate. As an argument to the procedure, the initiator supplies a description indicating that an immediate transfer is desired. ***Produce*** or *Consume* simply transfers the bulk data and returns. If the transfer fails for some reason, the procedure reports the error as shown in Fig. 4-5.

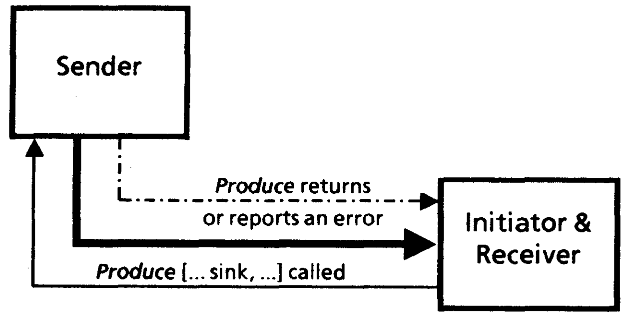


Figure 4-5 **Immediate bulk data transfer (initiator the receiver)**

5. APPLICATION SUPPORT ENVIRONMENT

A number of important support services are required for the secure, reliable, and smooth functioning of applications in a distributed network environment. These services are often not visible to the user but they are used by nearly all the XNS application programs. Together, they constitute the application support environment base. XNS implements the application layer (Layer 7 of the ISO model) in a structure which embodies a series of interrelated applications that build upon this support environment and use standard information formats. It is this richness of applications built upon a solid foundation that makes the XNS architecture so useful and versatile. The application support environment functions include:

- Providing the means for a user to locate specific resources or individuals on the network.
- Ensuring that only authorized individuals have access to sensitive processes and files.
- Establishing a common time base for use throughout the (potentially worldwide) network.
- Providing a common character encoding system, standard use of fonts, and font services.

These support functions are used by the higher-level application services discussed in subsequent sections and are available to XNS users for user-written applications.

Clearinghouse

One of the problems that must be overcome in a distributed system is identifying the location of system resources and users. If a workstation does not know how to gain access to a printer attached elsewhere in the network, it can't get something printed. If another workstation cannot locate a communication service, it can't gain access to the mainframe connected to that service. To communicate with other users, their electronic mailbox addresses need to be known. The problem is made worse by the constant changes occurring in networks (adding or deleting resources from the network, removing failed elements from service, dealing with elements that lack the appropriate attributes to perform specialized types of work, etc.). For a large network, keeping track of addresses and key attributes of system elements is a major undertaking. The Clearinghouse Service is conceptually similar to a telephone directory service, but with more powerful capabilities.

Object names and addresses

In XNS this problem is solved by Clearinghouse, a protocol whose purpose is to provide clients with the addresses of important objects (including people's mailboxes). These addresses are used in the remote procedures through which clients get work done.

Clearinghouse is essentially a data base of objects. The entry for each object consists of a name and a set of one or more groups of data items that encode the object's properties. Clients use the Clearinghouse service by providing it with object names and properties, in return for which Clearinghouse provides the appropriate address information.

Why are names needed at all? Why not just refer to an object by its address? Why not just directly use the network address of a file server, mail server, or printer? The reasons are much like those for using names in the telephone system or in a file system. First, address numbers are unintuitive; we do not want to refer to a printer by its network address any more than we want to refer to a colleague by a telephone number. Second, distributed objects change locations much more frequently than they change names.

Objects in Clearinghouse are named unambiguously in a uniform manner with the same naming convention for every object regardless of whether it is a user, a workstation, a server, a distribution list, or whatever. The naming is in a three-level heirarchy, and of the form: LocalName:Domain:Organization.

This division into organizations and domains within organizations is a logical rather than a physical division. An organization will typically be a corporate entity which can choose domain names to reflect administrative, geographical, functional, or any other type of divisions. Very large corporations may choose to use several organization names if their name space is extremely large. In any case, the fact that two addressable objects have names in the same domain or organization does not necessarily imply that they are physically close.

The Clearinghouse naming convention allows great flexibility and permits a user's localname to be chosen as his legal name (or a name of their choice, instead of some computer selected name such as DJones or DMJ). The Clearinghouse also supports the use of aliases and default names within the same domain for user convenience.

Mapping and binding

The Clearinghouse maps each name into a set of any number of properties to be associated with the name. A property consists of a Property Name, Property Type, and a Property Value. There are two types of property values: Type item which is an uninterpreted block of data, and the type group which is a set of names. Thus mapping a name into a network address is an example of a type item mapping, as in:

Daisy:SDD:Xerox → {<Printer, item, network address of the printer named Daisy>}.

A distribution list in electronic mail is an example of a mapping of type group, as in:

Authors:SDD:Xerox → {<Distribution List, group, {"Author 1:SDD:Xerox", "Author 2:SDD:Xerox"}>}.

Many properties may be associated with a name, as in:

John D. Smith:SDD:Xerox → {
<User, item, descriptive comment such as V.P. Marketing>,
<Password, item, password to be used for user authentication>,
<File Server Name, item, name of file server containing user's files>,
<Mailbox, item, name of mail server where user's mail is stored>,
<Printer Names, group, set of names of local printers any of which may be used>}.

In this example, the Clearinghouse is used to store the user's "profile." Note that the user's name was mapped into the *name* of his local file server (and mailbox and printer) rather than directly into its network address. This is because the name of the file server will perhaps never change but its location will occasionally change. We would not want a change in a server's location to require a major update of the Clearinghouse's database.

When a network object is referred to by name, the name must be bound to the address of the object. The later a system binds names, the more gracefully it can react to changes in the environment. If client software binds names statically, the software must be updated whenever the environment changes. On the other hand, binding takes time. Static or early binding increases runtime efficiency since, with either, names are already bound at runtime. A useful compromise combines early and late binding, giving the performance and reliability of the former and the flexiblity of the latter. XNS clients generally use early binding wherever possible and late binding only if any of these (early) bindings becomes invalid. Thus, software supporting printing stores the addresses of print servers at initialization and updates these addresses only if they become invalid.

The Clearinghouse naming and property declaration provisions are enormously flexible, making it possible for the most specialized aspects of a resource to be encoded in the Clearinghouse data base for future reference by clients. Clearinghouse also provides for pattern searches, in which a client can specify an object by indicating only those properties of interest to it. Clearinghouse will then ignore the other properties associated with the object in its database.

Generic names and Yellow Pages

The set of property names known to the Clearinghouse enables it to provide a Yellow Pages-like service. Client software can request a service in a standardized fashion, and need not remember what named resources are available. For example, each user workstation generally has a piece of software that replies to the user command "Help!" Suppose the generic name "Help Service" is agreed upon as the standard property name for such a service. To find the addresses of the servers providing help to users in domain:organization, the workstation software call asks to list all objects of name *"*:domain:organization"* with propertyname *Help Service*. The wildcard character "*" matches zero or more characters. This can be used by any workstation regardless of its location.

The "wildcard" feature allows clients to find valid names where they have only partial information or can only guess the name. It is particularly useful in electronic mail and in other uses of user names. If looking up *"Smith"* with propertyname *Mailbox* fails, because "Smith" is ambiguous, the electronic mail system may choose to list all names of the form *"*Smith*"*. with propertyname *User* to find the set of user names matching this name. It presents this set to the sender of the mail and allows him to choose which unambiguous name is appropriate.

Database replication

In all except the very small systems, the Clearinghouse (and its associated database) is decentralized and replicated. That is, instead of one global Clearinghouse, there are many Clearinghouse servers scattered throughout the internet (perhaps, but not necessarily, one per local network), each storing a copy of a portion of the global database. Decentralization and replication increase efficiency (it is faster to access a Clearinghouse server physically nearby), security (each organization can control access to its own Clearinghouse servers) and reliability (if one Clearinghouse server is down, perhaps another can respond to a request). However, conceptually it is assumed that there is one global database and each Clearinghouse server contains a portion of this database. No assumptions are made about how much of the database any particular Clearinghouse server stores. The union of all the local databases stored by the Clearinghouse servers is the global database.

The architecture allows a high degree of flexibility in what these distributed data bases contain. These can range all the way from single domain databases to the full global data base. The more domains a local Clearinghouse database contains, the higher would be the speed of response, because typically the local queries are answered faster than non-local queries. The updating of the multiple Clearinghouse databases is also automatic in the network. When a domain Clearinghouse updates its own domain data base, it also propogates the update if the database for this domain is replicated on more than one server. This update is done via electronic mail messages (which are time-stamped) so it is possible to have servers with temporarily inconsistent databases. The XNS architecture permits this; the Clearinghouse resolves these issues satisfactorily in the context of the services it provides.

Because many of the Clearinghouse exchanges are routine, they are often accomplished using the simpler Internet Packet Exchange Protocol, rather than the Sequenced Packet Protocol which, though more reliable, is more time-consuming.

The preceding discussion identified the Clearinghouse service as a separate system element as pictured in Fig. 5-1. While this may usually be so, on a small network the Clearinghouse can co-exist with other services on a single system server. (Such servers are called multifunction servers.) The Clearinghouse server is also a key contributor to network security in providing authentication and access control, the subject of the next section's discussion.

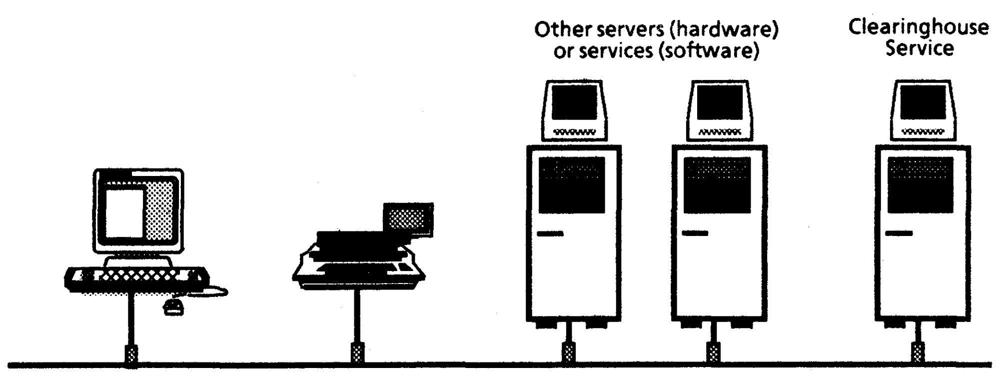


Figure 5-1 **The Clearinghouse Service**

Authentication

One of the difficulties with an open, distributed system is that access to system resources (files, printing facilities, etc.) is easy for any network user to obtain. Most of the time this is a distinct asset, but sometimes circumstances require that only certain users be able to gain access to a specific resource.

In XNS this problem is addressed by two provisions: access control mechanisms designed into appropriate workstation and services (e.g., filing, printing), and an Authentication Protocol which helps clients and services determine each other's identity in a reliable and secure way.

The solution provided by the Authentication Protocol assumes a secure Authentication Service which all clients and services trust to know their specially encrypted passwords, and that the clocks in the system elements are reasonably well synchronized. The Authentication Protocol provides for both a strong and a simple level of security. The goal of strong authentication is to make it practically impossible for one user to impersonate another, whereas simple authentication merely makes it difficult.

Password encryption

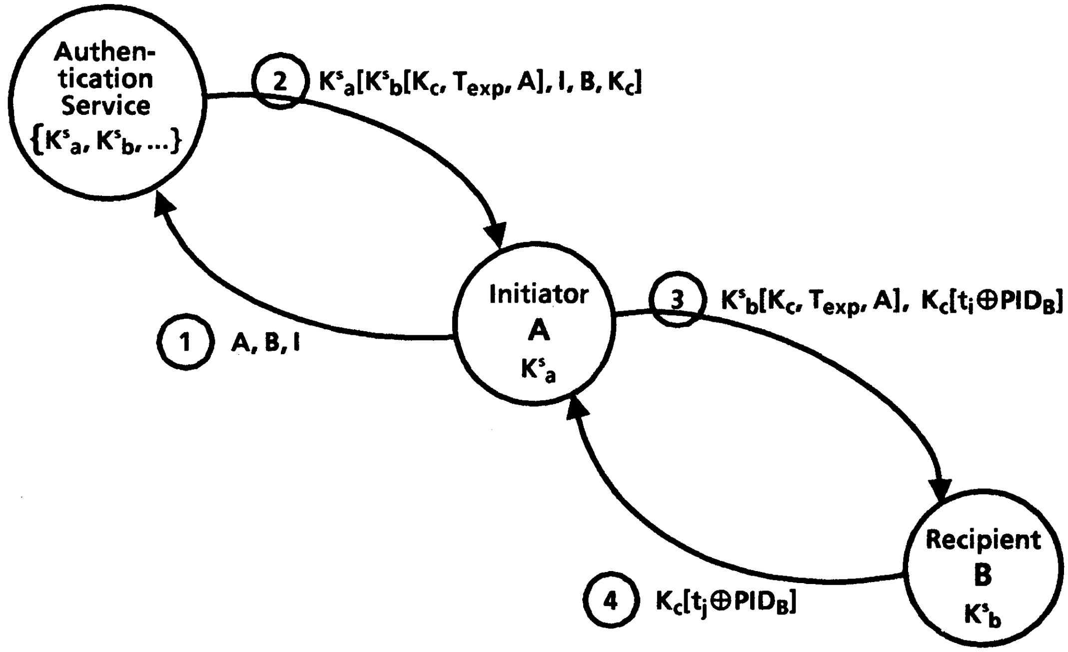
Every user has two passwords, strong and simple (a service has only a strong). The password used depends on the workstation encryption capability and the security environment. Passwords are for human users to identify themselves to the system. When entered into a workstation, the passwords are immediately encrypted according to a specific algorithim (hashing for simple, and NBS Data Encryption Standard for strong) to form a strong key or a simple key, thus ensuring that a user's password is never transmitted unencrypted.

Strong authentication

In slightly simplified terms, the strong authentication works in the following way:

Assuming that Party A (the initiator) wishes to identify itself to Party B (the recipient), Party A first uses the Authentication Protocol to contact a server on which an Authentication Service has been implemented, telling it the names of both parties and supplying a random number called the nonce (Step 1, see Fig. 5-2). The Authentication Service returns several pieces of information to Party A, all of it encrypted so only Party A can decrypt it (since the Authentication Service knows Party A's encryption key). Included in the return are the nonce, the name of Party B, a conversation key, and another set of information called credentials, which are further encrypted so that only Party B can decrypt them (Step 2).

Party A checks the nonce and Party B's name, comparing it to what had been sent. If it is the same, Party A can be reasonably certain the transaction is not a replay of a previous transaction. The credentials, encrypted with Party B's key, contain the conversation key, the name of Party A, and an expiration time that defines how long the credentials will be valid. The conversation key (which was also sent encrypted with Party A's key) is used to create a verifier by encrypting a time value (derived from the Time Service by exclusive OR of time stamp



Legend

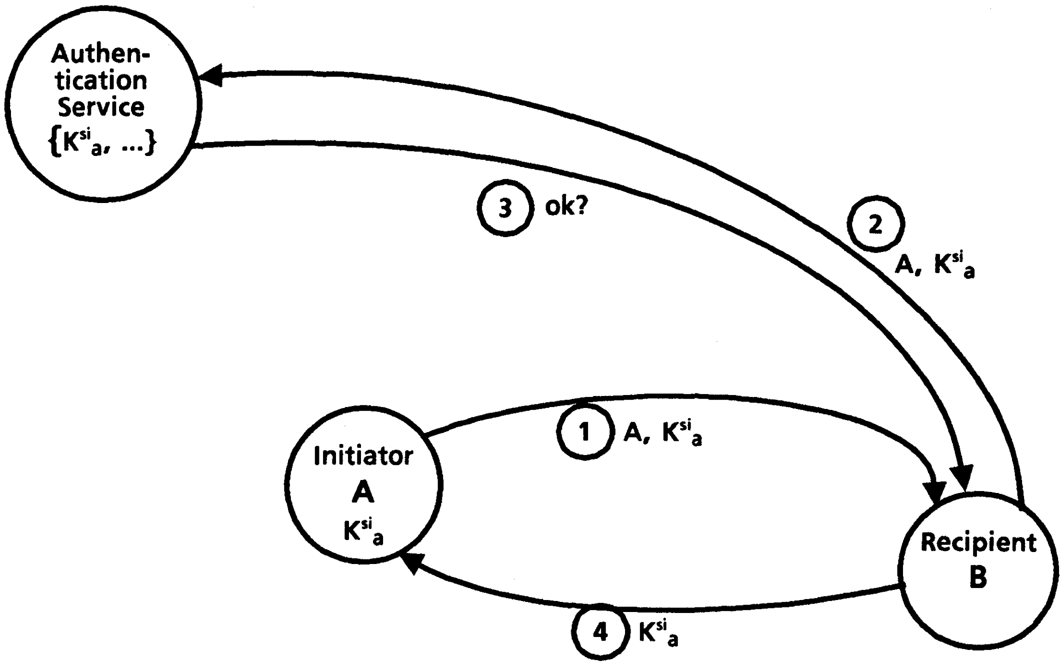
A, B	=	the fully qualified names of the initiator, **A**, and recipient, **B**
I	=	the nonce
K^s_a, K^s_b	=	the strong keys of **A** and **B**, respectively
K_c	=	the conversation key
T_{exp}	=	the expiration time for a set of credentials
$K_y[x]$	=	the value x encrypted with the key K_y
t_i, t_j	=	a time stamp obtained from the system clock at time i, j
PID_X	=	48-bit processor ID of X, padded after the least significant bit with 16 bits of zero

Figure 5-2 **Strong authentication model**

with recipient processor ID). Party A sends this verifier, along with the encrypted credentials originally supplied by the Authentication Service, to Party B (Step 3). Party B decrypts the time value and, by comparing it to the current time, can ascertain that the transaction is not an old one, and that the credentials (which identify Party A) can be trusted. Finally, Party B returns a verifier to A to ensure that Party B is not an imposter (Step 4).

Simple authentication

The Authentication Protocol also provides for a far simpler level of security. This acknowledges the fact that not all system resources are properly equipped to implement the tightest possible security, nor is it necessary or economically reasonable for them to do so. Since simple authentication is encoded by hashing, an initiator requires no interaction with the Authentication Service to manufacture the simple credentials. The simple authentication process is shown in Fig. 5-3; the initiator A sends its name and simple key to the recipient B (Step 1) which verifies A's simple key with the Authentication Service to assure of A's identity (Steps 2 and 3). Unlike strong authentication, the simple authentication verifiers are always the same, which means that an eavesdropper may obtain a valid credential-verifier pair and pose as the initiator. Also, the returned verifier (Step 4) provides no assurance that the recipient B is not an imposter.



Legend

A	= the fully qualified name of the initiator, A
$K^{si}{}_{a}$	= A's simple key
ok?	= a boolean which indicates whether or not $K^{si}{}_{a}$ is A's hashed password

Figure 5-3 **Simple authentication model**

The Authentication Protocol also provides for a privileged user, one capable of managing other users' keys and undertaking certain sensitive administrative procedures.

There also exists an Authentication operation called BroadcastForServers, which is used to locate instances of the Authentication Service in the internet. It is invoked using the Packet Exchange Protocol. A broadcast is made on the designated network and each Authentication server on that network returns its network address.

The security provisions of the Authentication Protocol, coupled with the access control mechanisms designed into the appropriate resources, ensure the necessary protection while allowing XNS to retain its openness and availability of resources. The Authentication Service usually co-exists with the Clearinghouse Service on the same server hardware.

Time

Various components of a distributed system must frequently obtain the current time. For example, file services need to record the time when a file is read or written, electronic mail messages need time stamping, and time may be needed for authentication purposes. To facilitate the acquisition of such information in a reliable and unambiguous way with a world-wide scope, a single time protocol and time standard is used throughout Xerox Network Systems.

Time Standard

The Time Protocol uses the Time Standard which defines time as a Courier datatype, available for use by any of the Courier based remote procedures. Its standard representation is also used in the definition of other non-Courier based protocols. A data object of the type Time is a 32-bit number which represents the current time unambiguously in seconds. This representation gives the time starting from 12:00:00 AM (the beginning of the day), 1 January 1968. The 2^{32} seconds (about 136 years) represented by Time extend into the 22nd century before any ambiguity with a past date is encountered.

Time Protocol

The Time Protocol specifies the manner in which a Time Service makes the current time available to its clients on other system elements. (These clients may be workstations, terminals or servers.) It is built upon the Internet Packet Exchange Protocol. The response packet from the Time Service provides the current time, along with its time zone, and information about when Daylight Savings Time is observed at its location, as shown in Fig. 5-4. The additional information is for client convenience which may (but need not) use this information in formatting the time for human consumption.

This global approach to the handling of time is one of the reasons why XNS systems can be implemented uniformly across geographical boundaries which would restrict systems based on architectures designed with a less global perspective.

Character code and fonts

Information interchange on a world-wide internet requires a fundamental rethinking of the encoding of characters, the basic information element in written languages. In the United States and other English-speaking countries, the 7-bit ASCII (American Standard Code for Information Interchange) is widely used in all varieties of workstations, terminals, and computers. ASCII is capable of representing basic English upper- and lower-case letters, numerals, punctuation, com-

word 0	protocol version = 2
word 1	packet type = 2 (response)
word 2 word 3	current time
word 4	offset direction
word 5	offset hours
word 6	offset minutes
word 7	start of Daylight Savings Time
word 8	end of Daylight Savings Time
word 9	tolerance specified
word 10 word 11	tolerance
	← 16 bits →

Figure 5-4 **Layout of data field of Time Service response packet**

monly used graphic characters and communication control codes totalling 128 different characters.

But 128 characters are hardly enough to deal with the printing needs of English, let alone other languages. To correct this situation, ASCII has been extended to 8-bits to define an additional 128 characters. The International Standards Organization (ISO) has also adopted a similar 8-bit character code standard commonly referred to as ISO 646. These 8-bit character codes, while a vast improvement over 7-bit ASCII, are still inadequate. Special accents used in many European languages, Greek, mathematical symbols and printing-oriented characters quickly exceed the 256-character capacity of any 8-bit code. When the needs of the many different languages in the world are taken into consideration, any hope for using a restricted code space vanishes. Chinese and Japanese each have requirements for thousands of characters.

The ASCII and the ISO 646 standards, as well as the numerous other national and international character codes that are being used, permit only limited information interchange. However, it is vitally important to use these standards to communicate with the millions of workstations, terminals, and computers that use them. A global information system must intelligently deal with all of these standards in a uniform and unambiguous way, while retaining compatibility with existing codes.

Character code standard

The Xerox solution to this problem is a character encoding system which normally conforms to the ASCII and ISO 8-bit character codes, but expands to a 16-bit code when necessary. The 16-bit coding scheme permits 65,535 different character codes which is sufficient for encoding all of the commonly used human languages. However, should future requirements warrant, mechanisms exist to expand the character code space beyond 16-bits.

Multilingual capabilities

The April 1984, published version of the Character Code Standard (also known as the Xerox Character Code) specifies the character code assignments for Greek, Cyrillic, and Japanese characters in addition to the Latin character set defined by ISO 646. Additional characters are being added to this set for new technical domains, and the alphabets of more languages, including Arabic/Farsi, Hebrew, Hindi (Devanagri), Chinese, and Korean. Updates to the standard will be published periodically to reflect such additions. The fact that all the different languages can be conveniently represented in a single character code standard simplifies the design of multilingual document generation and printing systems. Such multilingual capabilities are available in many Xerox products. They allow a user to mix text in different languages, use any of the Greek symbols in mathematical equations, and put everything in the same electronic document in a uniform and consistent manner.

Character Code model

The Character Code Standard may be used worldwide for any number of applications including communication protocols, electronic printing, electronic mail, keyboard input, document editing and document interchange. The standard assigns a unique, unambiguous and absolute numerical code to each semantically different character to permit efficient storage and processing while also ensuring proper interpretation of information. To understand how this assignment is made, the 16-bit character code space may be viewed as a series of 256 blocks of 256 codes each. Each such block is called a character set. Each 16-bit character code may be viewed as consisting of two 8-bit bytes, the first of which is the character set code, and the second an 8-bit character code within that character set, as shown in Fig. 5-5. The character codes are assigned so that characters within a single character set tend to be related to each other by traditional usage. Thus, all of the characters in Character Set 0 are for the Latin set. Some alphabets, such as Japanese Kanji, which involves well over six thousand characters, require many character sets, as shown in Fig. 5-6.

High-quality typography

The Character Code Standard also allows for high quality electronic typography by providing rendering characters which are variant representations and special sequence of graphic characters, such as ligatures. Other entities, such as logos and

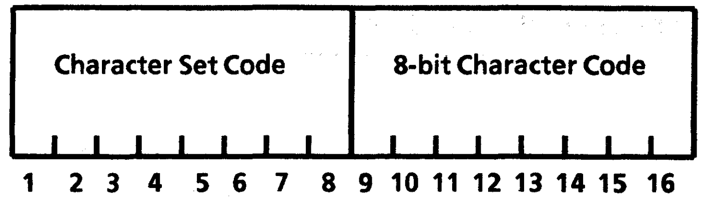


Figure 5-5 **16-bit character code**

people's signatures, are not defined by the standard. A portion of the rendering code is reserved for private use to accomplish their conversion into hard-copy on an electronic printer. These provisions make high quality typography easier on a system-wide basis.

Encoding efficiency

It would seem that 16-bit codes would take twice the storage and transmission time as 8-bit codes. Normally, this would be true, but the Character Code Standard also defines a string encoding standard which compresses the 16-bit codes into 8-bit bytes on a one-for-one basis (i.e., one 8-bit byte for each 16-bit entity), thus providing versatility with little or no loss in efficiency. Moreover, all sequences of 8-bit ISO 646 characters may be stored or transmitted as they are. For languages such as Japanese, which normally require the use of a 16-bit code, text may be stored and transmitted as a sequence of 16-bit codes.

Compatibility with other codes

Every effort has been made by Xerox to be compatible with the large number of national and international code standards. For example, the Character Set 0 assignment is fully compatible with the ISO 646 and the ASCII standards, and the Japanese Kanji assignment is in accordance with the Japanese Industrial Standard Code JIS-C-6226. However, it should be recognized that the various national and international standards are not consistent among themselves, so some compromises must be made for the goal of full compatibility. Nevertheless, the Xerox Character Code Standard represents one of the most consistent and successful international implementations available.

Font architecture

A character's numerical code expresses its identity, its content, and its semantics, but this does not fully determine a printed or displayed character's visual appearance. For example, the character with the same unique code can be printed in a different size, style or typeface, and look very different. The term font is used to define a particular collection of characters of a typeface with their unique orientation, size, posture, weight, and other attributes. Together, the character code and the font define the visual appearance of a character. A font architecture, defining font representations, font file formats, font names, and font services is important to assure that documents are displayed and printed with high quality and uniform consistency; that is, a document "looks" the same

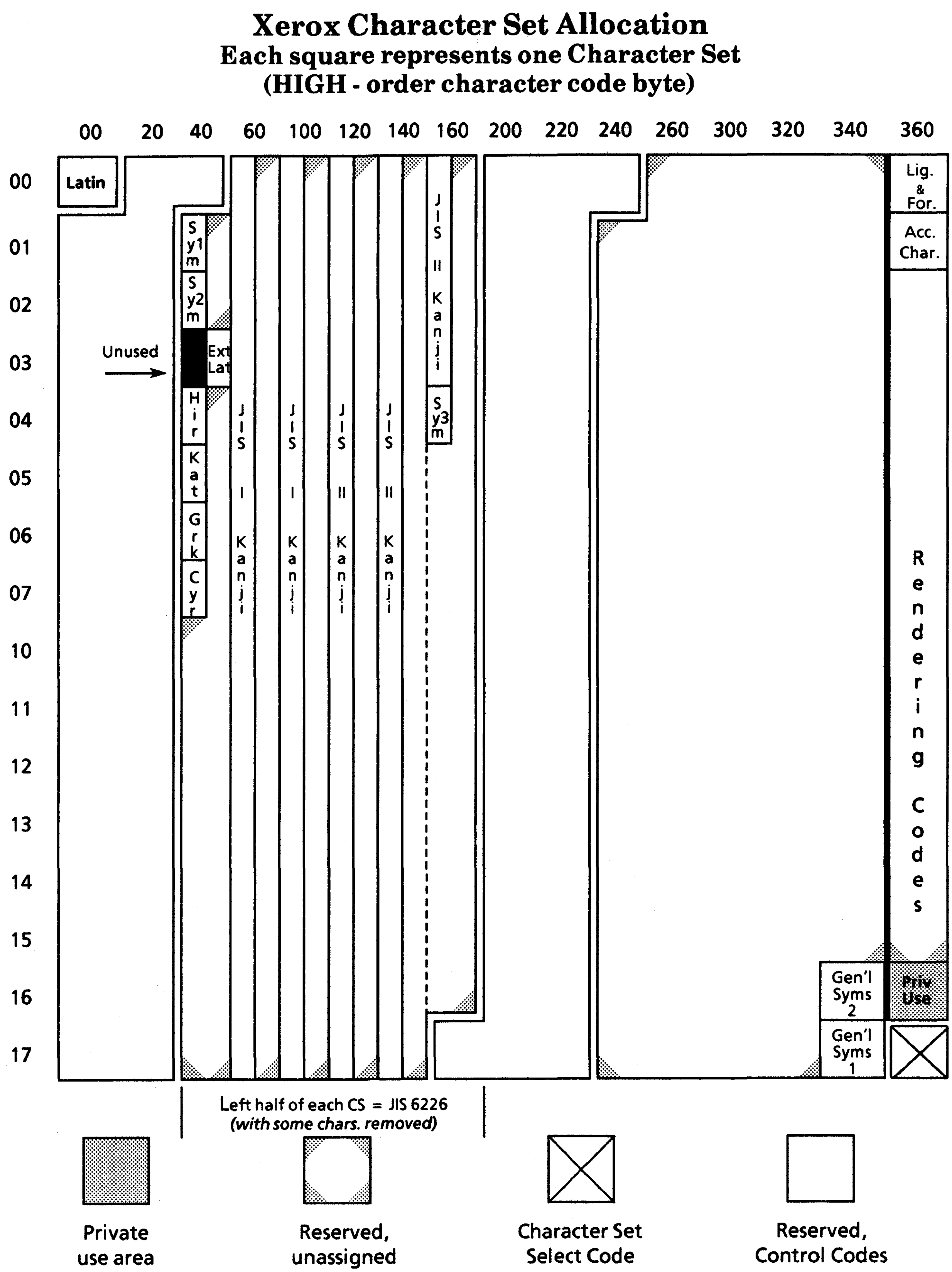


Figure 5-6 **Xerox Character Set allocation**

whether it is displayed on a workstation screen, typeset on a typesetter, or printed on any number of electronic laser printers available from Xerox and others.

Font representation

Fonts may be represented and stored (for later printing or display) in a number of different forms. Two of the most common forms of representation are bit map representation and contour representation. The bit map representation generally offers higher quality and greater processing efficiency, while the contour representation (often stored in spline form) generally offers greater flexibility with regard to choice of size, orientation, and resolution ("dots per inch"). The Xerox font architecture allows for both types of representations and gives the user the flexibility to choose the representation that best suits the needs of a particular application. Most Xerox products use the bit map font representation.

Font file format

A font file is a set of font records including the digital representation for the characters and control information for processing. A standard font file format provides for easy interchange of fonts between all font-using products (workstations, printers, etc.) and thus increases font consistency between products.

Font names

Xerox has standardized a font name structure for its own use which is defined in the Print Services Integration Standard. In this structure, used in Interpress, the first three identifiers of a font name vector uniquely identify a font. These identifiers are: Naming authority (Xerox, NBS, etc.), character code (e.g., Xerox Character Code, EBCDIC, etc.), and typeface (e.g., ItcGrammd-Demibold-Oblique-DesignSize9pt). Xerox also provides a Font Name Registry for those wishing to register their fonts with Xerox. The typeface identifiers themselves have no mandated structure, but guidelines are suggested to help in achieving greater uniformity.

Font service and protocols

A font service supplies printers, workstations, composers and other font using devices with fonts and/or font metrics upon request. A font service may also be able to initiate a distribution of fonts to font-using network devices. A font server is a network service which provides this function using a font protocol. A schematic of the use of a font service by workstations and printers is shown in Fig. 5-7. According to this model, an electronic printer or electronic printing system may request bitmap or contour fonts from the font service (Step 1A), which supplies these fonts from its database (Step 1B). The printer (or any other system element) can (if so desired) use the font service for archiving bitmap fonts (Step 1C), which it may have generated from contour fonts. Alternately, the font service may be used by a workstation to obtain screen fonts (Steps 2A and 2B). A composition system such as a mainframe or publishing system connected to the network can also use the font service to obtain the required font metrics (Steps 3A and

3B). The interactions with the font service use the font protocol, and fonts and metrics are transmitted using standard font file formats. The use of this model enables greater availability of fonts and consistency in their usage.

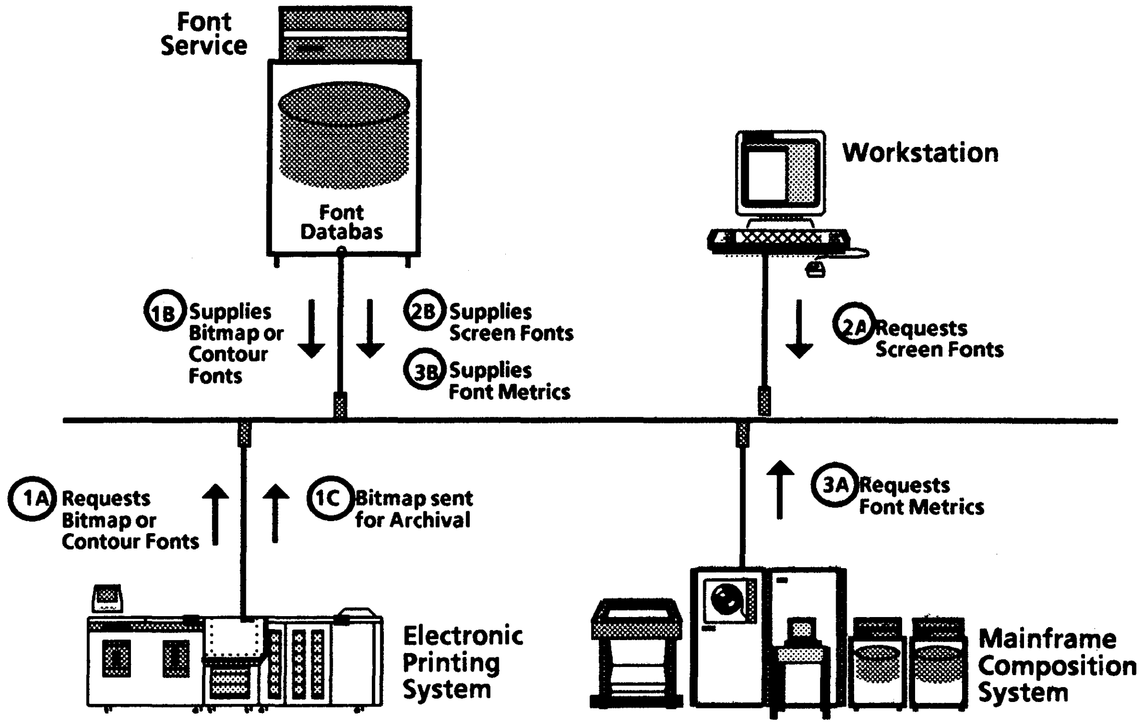


Figure 5-7 **A font service**

6. GATEWAY ACCESS

One of the more challenging problems in a network architecture is interfacing other systems that were designed according to different, incompatible architectural standards. Yet the problem must be solved because with the proliferation of all kinds of computers and networks, users are demanding the integration of diverse systems, including office information systems, conventional data processing systems, manufacturing and engineering information systems, and retail and financial transaction processing systems.

If one system is to integrate successfully with another, there must be a way to convert the communication protocols, document and data formats, file structures, commands, and control functions used by another system. In XNS, the responsibilities for these conversions are located in a gateway, a hardware/software subsystem intended to bridge the differences between two incompatible systems.

Recognizing that the problem of interconnecting with other systems will continue to arise indefinitely, the XNS designers chose to implement a variety of gateway services. These gateway services provide access to and from an XNS system for a variety of other systems and devices, including those that use TTY, VT-100, and the IBM 3270 protocols. The Gateway Access Protocol is used in communicating with other non-XNS systems.

Gateway Access Protocol

The basic functions performed by the Gateway Access Protocol (GAP) are to move information, support other communication models, and support terminals.

Communication support

GAP supports many communication models, including the document transfer model for electronic mail, transaction-oriented model for remote data base access, and an interactive model for interface to a mainframe data processor. To achieve this, GAP provides flexible control of the unit of data transfer and the frequency of transmission activity. GAP does not, however, deal with the content of data which is application dependent. It provides *information transcription* but **not** *information translation*. Information translation may be provided by either party using GAP, or via a separate conversion service, such as a Document Interchange Service in gateway or workstation products. In supporting other communication models, GAP adapts to the other protocols when different from XNS. Thus many of the commonly used communication protocols are supported by the XNS gateway services.

The Gateway Access Protocol (GAP) is an application-level protocol that makes use of the Courier and Internet Protocols (specifically, Sequenced Packet Protocol) to interconnect an XNS system with a non-XNS system or device. GAP uses these protocols to issue or receive customized command and data sequences that exactly replicate the command and data sequences used by the target system or device. Courier is used to establish a session with the target and the Sequenced Packet Protocol is used to transmit and receive the appropriate bit and character sequences.

Virtual terminal support

The GAP creates a logical appearance of communication compatibility to the other system or device. For interconnecting with a remote mainframe system, this appearance emulates particular devices, such as TTY, VT-100, or IBM 3270 terminals, which the mainframe supports. GAP also makes it possible for non-XNS terminal devices to interconnect with an XNS system and access XNS services. For this reason, the Gateway Access Protocol is also referred to as the Virtual Terminal Protocol (VTP)

Gateway services

In order to integrate successfully with other devices and systems, XNS is equipped with a variety of gateway services that use the Gateway Access Protocol. For example, the External Communications Service is used for integration with mainframe computers and the Interactive Terminal Service provides access to non-network terminals.

Fig. 6-1 illustrates how a communication server is used to make the physical and logical connections to a remote terminal and to a large mainframe. A workstation on the Ethernet to which the server is connected is then capable of creating documents or records and sending them to the mainframe and/or accessing information stored at the mainframe. The remote terminal can input document content to the XNS system for subsequent editing, storage, or printing; it can also send and receive mail. The communication server can initiate and maintain many simultaneous connections between the internet and the remote systems. A Communication Interface Unit may also be used in conjunction with the communication server to provide additional communication ports.

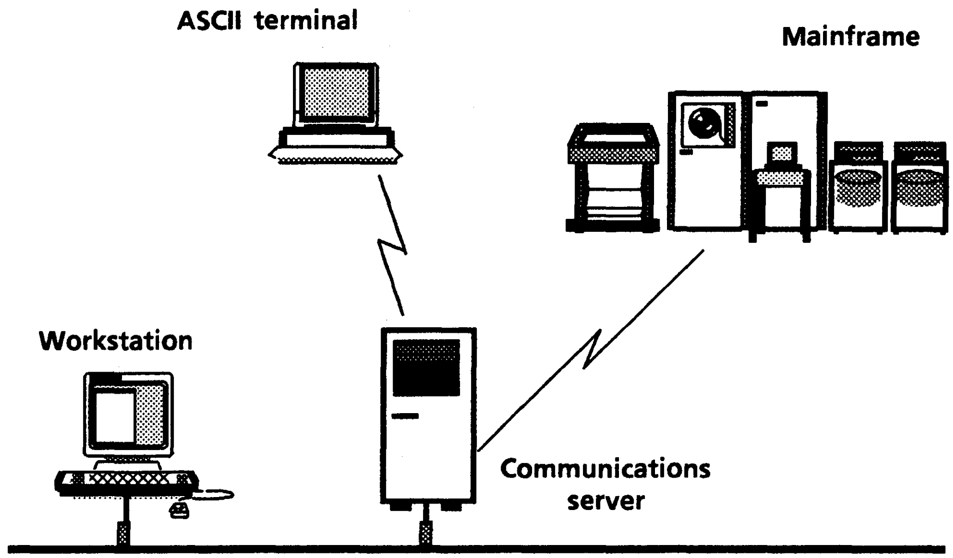


Figure 6-1 **Gateway access to/from non-XNS systems**

The many different types of gateway services are generally implemented as separate software products which may be co-resident on the same server or have separate server hardware.

External Communication Service

Access to mainframe computers

XNS workstations use the External Communication Service (ECS) to access non-XNS mainframe computers. This service works in concert with terminal emulation capabilities in Xerox workstations which simulate the appropriate terminal protocol. To support these terminal emulation capabilities, the ECS performs three functions: It communicates with the host in the host's native protocol; it communicates with an XNS workstation in XNS protocols; and it provides the sharing of communication resources to connect the host to all XNS workstations.

Terminal emulation in XNS

When a workstation user requests an emulation session with a host computer, the workstation uses the Clearinghouse to locate an External Communication Service that supports connections with that particular host. The workstation then connects to the ECS which initiates a session with the remote host and performs conversions between XNS and the remote host protocols. The workstation presents a user interface equivalent to that of the terminal being emulated. The protocol conversion provided by the ECS allows information originating either in the mainframe computer or anywhere on the XNS internet to be transferred and integrated into the mainframe's environment for use in various applications.

A workstation is not restricted to using only local External Communication Services. Since the XNS protocols support communication anywhere on the internet, a workstation that supports the XNS protocols can use any ECS on the internet.

IBM 3270 emulation

Major portions of IBM's extensive repertoire of mainframe computer software are designed to work with IBM's 3270 display terminal systems. Hundreds of thousands of these terminals, and various plug-compatible equivalents, have been manufactured in both clustered and standalone versions.

During IBM 3270 emulation, the ECS supports the IBM 3270 Binary Synchronous Communication (BSC) or the IBM 3270 System Network Architecture (SNA) protocols. An IBM 3276-2 cluster controller is emulated when using BSC protocol and an IBM 3276-12 cluster controller is emulated when using SNA protocol.

Like the IBM 3276 cluster controllers, the ECS can support up to eight concurrent emulation sessions. Unlike IBM cluster controllers which require their attached devices to be within the vicinity of the controller, the XNS emulation sessions may originate either from a single workstation or from several workstations, anywhere on a worldwide internet. Software options enabled at the XNS workstations allow their users to interact with an IBM host computer in an IBM 3270 emulation mode. This feature is also available for the non-IBM mainframe and large minicomputer systems which are compatible with the 3270 terminals. Thus by emulating the IBM 3270 terminal, an XNS workstation can gain access to a very large number of mainframe computers.

VT-100 and TTY emulation

The ECS also contains the software option that supports emulation of VT-100s and TTY-type terminals. TTY emulation is generally via a physical dial-up connection for each session. The protocol used between ECS and the host is Asynchronous ASCII. Workstations emulating teletype (TTY) terminals present interactive displays that are equivalent to the standard teletype terminal. The ECS maintains the bi-directional communications necessary for the emulation session by transforming the ASCII protocols into XNS protocols, and vice versa. This capability makes it possible to interconnect XNS workstations with most non-IBM general-purpose computers.

Support of incoming calls

The ECS also supports incoming calls from personal computers or TTY-type terminals that are not part of XNS. Users can access the Interactive Terminal Service through a dial-up connection to an ECS port.

Interactive Terminal Service

The Interactive Terminal Service (ITS) allows users of workstations and terminals not directly connected to the XNS internet to access XNS print, file, mail, and other resources. Any personal computer or standard ASCII terminal that can initiate a dial-up communication session using the asynchronous protocols can communicate with the ITS and use the network services on the internet. Nearly all computers with standard RS-232-C communication ports are able to operate that way. This includes Xerox Memorywriters, personal computers, professional workstations, and text processing systems.

ITS provides an interactive user interface to give users access to XNS resources. Users at remote devices interact with this user interface to store and retrieve documents on a file service, print documents on electronic printers, or send and receive electronic mail. During a communication session a user has the illusion of being directly coupled to the internet. That's because ITS communicates with the internet using XNS protocols once it determines what a user wants to do.

ITS document interchange

To enhance the exchange of messages and documents between terminals and workstations, ITS supports a variety of document formats including 860 document format and plain text.

If the intended recipient is a workstation on the internet, ITS can be instructed to convert the document into the format suited for that workstation.

ITS mailing

After the user establishes a connection and logs on to the ITS, he or she can issue commands through the mailing interface and request various mail operations. Messages can be composed and sent to different individuals and distribution lists, or the user can retrieve and edit the messages received.

Messages are not limited to text entered while the user is connected to the ITS. The user can include a previously prepared document in a message. This means a user can work on a document and send it over the phone network to anyone who has direct or indirect access to the internet. When a document is sent to a Xerox workstation, the recipient can improve it with the workstation's advanced text and graphics capabilities and then print, file, or mail the document to other users on the internet.

ITS filing

ITS also supports file transfer between non-XNS devices and the File Service, using the XModem asynchronous communication protocol. Users can create files in any format, using their own word processing or spreadsheet software. These files can then be transmitted from the device to ITS and stored in their original format on the File Service. The stored files can be displayed and edited by XNS as well as non-XNS devices.The storage and retrieval operations using the XModem protocol allow output from multi-vendor computing devices to be shared on the network.

ITS printing ITS supports the use of XNS printing for documents in Interpress, the Xerox electronic printing standard. Conversion to Interpress is also available for a limited set of formats.

850/860 Gateway Service

The 850/860 Gateway Service allows users of non-networked communicating Xerox 850 and 860 text processors and the Kurzweil Intelligent Scanning Systems, to exchange information over telephone lines with users on the internet. By dialing the 850/860 Gateway Service, non-networked users can exchange documents with users of workstations on the internet, or with other remote workstations. This flexible extension of the internet is particularly valuable in organizations with 850, 860, or compatible workstations located in small branches of the organization.

The Gateway Service communicates in both 850/860 and XNS protocols. Because XNS can handle information in 860 format, the 850/860 Gateway Service acts as a transfer agent between the 860 and the internet while preserving the enhancements in the original document. If documents originate at an 850, the 850/860 Gateway Service converts the 850 format to 860 format before making the transfer.

Remote Batch Service

The Remote Batch Service (RBS) enables XNS users to exchange documents and transfer files with devices that implement the IBM 2770, 2780 and 3780 Binary Synchronous Communication protocols used by IBM Remote Batch Terminals, as well as by word processing and data processing terminals that emulate them.

The RBS basically performs the following tasks: It translates between document formats during the interchange of documents, providing compatibility with XNS; it acts as a third-party transfer agent for formats which it does not understand; and it transmits XNS files to a mainframe computer for storage and archiving without any loss of structure or information.

The RBS can be used for document interchange, document transfer, or document archiving applications with a mainframe. Tasks sent to a mainframe require an additional document that specifies the Job Control Language to execute the task on the mainframe.

7. FILING

Word processing equipment and most personal computers store documents in electronic form on flexible diskettes holding several hundred thousand characters. Since diskettes can be removed from a machine and stored off line, there is almost no limit to the number of characters that can be recorded this way. But storing documents on diskettes is inconvenient when they must be shared with others, especially those in remote locations. And off-line storage can be troublesome if no one can locate a particular diskette.

Documents stored locally at an XNS workstation may also be stored remotely in a file service. High-capacity disk files are able to store documents to make them available to other authorized users on the internet. This internet-wide file sharing capability provides many benefits to a user such as access to the most up-to-date information, ability to work jointly on a document even from remote locations, and creating document databases pertaining to activities of a small group or an entire organization.

File service requests

A file service deals with action requests from two different kinds of sources. The first represents the user who wishes to access the file server. This access must be provided via a protocol; in this case, the Filing Protocol, which is layered above Courier, as are all XNS applications protocols.

The second source of action requests comes from system administrators, persons who desire information about the current state of the File Service or who wish to make administrative changes (e.g., creating a new file drawer) authorized by their credentials. System administrators may perform their actions through a local TTY terminal or remotely via the Gateway Access Protocol, also layered above Courier. System backup operations are initiated through this interface as are many operations which authorize system access. These back-up and archival features of a file service are generally not visible to the client, and are not specified in the Filing Protocol.

Filing Protocol

In XNS, the interaction between clients and file services is defined by the Filing Protocol. The Filing Protocol is both a guide for using a file service and a specification for the implementation of such a service. It is not a description of a particular implementation of the protocol. This protocol provides a general filing facility to support a wide variety of applications. However, it is not intended to directly support network administration functions, printing, electronic mail, or other distributed activities. These are subjects of other specifications.

Filing Protocol model

The Filing Protocol follows a session-oriented model in which a client interacts on behalf of a user (which may be a human or other entity such as another service); and must log on before using the file service. If the log on is successful, the service establishes a session during which the client interacts with the service. When interaction is complete, the client logs off to terminate the session. The client must log on again before any further interaction may occur.

Sessions may vary greatly in duration. In some patterns of use, a session is established to perform a single operation and then terminated. In others, a session may last a very long time even though it is largely inactive. There may be several sessions simultaneously in existence for the same user whether or not they were established by the same client. The file service reserves the right to terminate a session at any time that a remote procedure call is not in progress. This might occur if a session remains inactive for a long period or if the system element supporting the file service has to be shut down.

The file service stores and operates on files. It does not control the content of what it stores. Documents in any format may be stored in or retrieved from a file service as shown in Fig. 7-1. If a document in ASCII format is stored and then later retrieved by a non-ASCII device, the document will essentially be unintelligible. A document conversion service must be used to convert information encoded in one format into the format used by a dissimilar device.

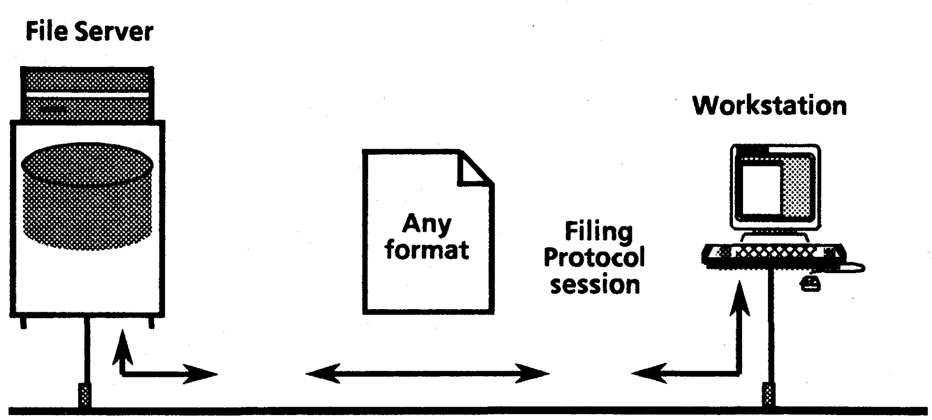


Figure 7-1 **Transmitting documents between a workstation and a file service**

Filing Protocol structure

The Filing Protocol may be viewed as being composed of four sub-layers, one above the other, which progressively provides additional functionality (see Fig. 7-2). These four sub-layers, from lowest to highest, are:

1) Session; 2) File; 3) Directory; 4) Search

SEARCH **locating, "wild-card" matching**
DIRECTORY **hierarchy, sub-directories, move, copy, delete** **attributes and access controls**
FILE **open, close, handles, file** **attributes and access controls**
SESSION **log-on, authentication, interlocks, log-off**

Figure 7-2 **Filing Protocol structure**

Session sub-layer

The session sub-layer is responsible for providing a context within which a sequenced set of related action requests from the client can take place.

It begins with a process which authenticates to the File Service that the client is who he claims to be and which authenticates to the client that the service is indeed the one intended (to prevent impersonation of a service). This process utilizes the Authentication Service via the Authentication Protocol. Encryption (DES) is utilized to make this authentication process secure.

The authentication determined at the beginning of the session is subsequently utilized to gain permission to access files.

The session also provides a context for a system of software interlocks which coordinate the access of two sessions competing for access to the same information. Without such interlocks, shared information can easily become inconsistent.

File sub-layer

The files sub-layer is responsible for the *file* data structure and, within the context of a session, for implementing those operations which deal with the individual file. The Open and Close commands provide a handle (an identifier used in client-service interaction) on a file within the context of the session.

A file consists of content and attributes. The content is a sequence of octets (an octet is a group of 8 bits of binary information not necessarily representing a graphic character).

The File Service does not attempt to place any interpretation upon the octets of a file's content (with the exception of when the file is a directory, as explained below for the directory sub-layer). The Retrieve and Replace commands transfer all or part of the content of a file.

Attributes

Attributes are additional information about the file: they are partitioned into interpreted and uninterpreted attributes. Uninterpreted attributes are only for the use of the client and are not processed or acted upon by the File Service other than to record them and to return them to the client. The interpreted attributes, on the other hand, have meaning to the File Service and are, in general, maintained by the File Service. While interpreted attributes are all defined at the files sub-layer, some are maintained by the files sub-layer and others by the directory sub-layer.

At the files sub-layer, some of the interpreted attributes are:

— a binary file identifier
— a file name as a human sensible character string.
— a flag indicating temporary or permanent
— a version number
— a checksum
— the file size (in octets)
— access control information
— the time and identity of the client performing the operation.

Directory sub-layer

The directory sub-layer organizes the files into a hierarchical tree as is customary. Some additional attributes do the actual function of binding the files together into the tree structure.

Many commands deal with a whole sub-tree, manipulating a directory and all of its files including its sub-directories at all levels. These commands include Move, Copy, and Delete. These commands are extremely powerful because of the large scale reorganization of the files which a single command can invoke.

Search sub-layer

Finally, a Set command exists which can identify a whole set of files related by similarities in their human understandable file names. This set is identified by providing a file name-like pattern containing various "wild card" characters. The search sub-layer generates a list of all files whose name matches the pattern and provides this list to the client, who is free to do what he chooses with it.

This mechanism is often utilized by a user interface to support a "see and point" mechanism of file identification. By displaying a list of candidates, a user can select the proper file.

Printer Subset of Filing

It is sometimes desirable to allow remote access to a printer's file system so that files may be stored there and used at a later time. Forms overlays, masters to be referenced from other masters, and font files are examples of objects that a client may want to store on a printer. The Printer Subset of the Filing

Protocol is designed so it can be implemented as an adjunct to a Print Service. Thus, it provides much more limited functionality than the Filing Protocol. It is a strict subset of the Filing Protocol. This means any behavior defined in this subset protocol is identical to the behavior defined in the Filing Protocol. Clients of this protocol are guaranteed that they may talk to services implementing the full Filing Protocol simply by using a different service identifier.

Data transfer

For those filing procedures that intrinsically require the transmission of a large amount of data, the Bulk Data Transfer Protocol is employed. Basically it works as follows: Between the call to a remote procedure and the return from that procedure, the sender (either client or service) uses the bulk data transfer mechanism to send to the receiver the attributes or the content of the designated file(s). Note that the Bulk Data Transfer Protocol (with its third party transfer) allows the data to be sent to, or retrieved from, a system element different than that of the client.

File Service

The XNS file services give users access to shared files anywhere on the internet. This sharing of information is not limited to XNS workstations. Any authorized user may access the file service from any terminal, personal computer or even an electronic typewriter (such as the Xerox Memorywriter) which has the required communications interface. This access is enabled via the Interactive Terminal Service.

File organization

File sharing is made easy by the organization of the file service. Files in the file service are organized hierarchically; a major directory (file drawer) at the top can contain several sub-directories (folders), each of which can contain additional layers of sub-directories or documents (record files, spreadsheets, mail notes, etc.) with no limit to the number of descendant directories. (There is, however, a limit on both the file drawer size and the total amount of storage available on the file service.)

All file drawers, folders, and documents are given names, and all levels of hierarchy contain the names that are descendants of the preceding level. The file service also attaches a unique identifier to each file which indicates its position in the hierarchy and provides efficient access to the files.

In addition to a name, the file service associates a number of attributes with each object in the hierarchy. Attributes describe the current characteristics of a file and help to distinguish it from other files, but are not part of its content. Examples of attributes are the date/time stamp associated with the file at the time it is entered into the file service, the name of the creator, the access privileges associated with one or more individuals or groups and the number of pages allotted to the file drawer. These attributes are used by the file service to provide a number of benefits to the user. For example, files can be sorted in a folder by creation date or by name according to the user's needs.

The file service automatically sorts the contents of folders according to the attributes established for each. For example, files in a folder might appear in descending alphabetical order by name or in numeric order. A user can sort by ascending or descending alphabetical order, or by creation date, or have the folder unsorted. This flexibility makes it easy for users to organize and locate a particular folder or document.

Access controls

Since XNS is a distributed system (allowing users to access all services across the internet), access controls must be available to protect private or restricted information. The File Service allows access controls to be placed on file drawers. These controls determine who is permitted access and the type of access permitted.

Each file drawer has an owner registered in the Clearinghouse. The owner has complete access to the file drawer. The system administrator or the owner establishes access rights for other users to each file drawer. Only users listed in the file drawer's access list can read or alter a file. The owner can designate that some users be given only partial access rights (such as the right to read files), or more extensive rights (such as the right to delete, move, or add new files). Any person not listed will be denied all access to the file drawer and its contents. The controls placed on the file drawer apply to all of their descendant folders and files.

File service back-up

A regular back-up of the contents of a network file service is essential. Backup is a procedure in which the file service contents are copied onto other media or to another file service. This backup occurs at intervals determined by the system administrator and can take place unattended. If information is lost or accidentally deleted, it can be restored from the backup copy.

8. DOCUMENT INTERCHANGE

Document compatibility

Paper documents exist in many forms and so do electronic documents. Document-oriented information systems are typically made up of various kinds of editing devices such as word processors, advanced workstations, terminals of different kinds, and automated composition systems. Extensive distributed information systems also frequently provide interconnection to external devices and systems which are themselves capable of document creation and editing. This type of information system risks accumulating documents created by a wide variety of devices over a long period of time. Although such documents can be stored and retrieved by means of XNS Filing (in the case of XNS) without regard to their origin or content, any given document cannot normally be subsequently edited except ***by an editor compatible with the editor that initially created the document***.

To "edit" a document means both to modify it and include any unmodified portion of it in other documents. Usually, either of two incompatible editors is unable to perform any operation on the other's documents.

For new systems this situation is usually only a nuisance. But as systems grow, new editors are introduced, existing editors are modified to the point where they will no longer accept documents produced by earlier versions, and documents are introduced into the system through exterrnal communication links. As the demand for cross-editing grows, a situation that began as a nuisance will become intolerable.

The reasons that documents produced by different editors are incompatible include the use of different data codes, different structural conventions, different techniques for indicating special conditions such as underlined text, the provisions made for explicitly declaring output characteristics, etc.

One potential solution to this dilemma would be to design a translation service that directly converts the private form native to one editor into a form that can be understood by another editor. This has two drawbacks, however. For even a modest population of different editors, this suggests a potentially large number of different translation services. And whenever design changes are made to one editor, corresponding changes must be made to each translation service that understands that editor's format.

Interscript

The Xerox approach to this problem is to adopt a document encoding standard that is sufficiently general so any document can be described in its terms, and to provide a way to convert the private forms of idiosyncratic documents to/from this new encoding. The encoding standard is thus an interchange standard that exchanges (not edits) documents between editors.

The encoding standard designed by Xerox is called Interscript. Fig. 8-1 shows document interchange using Interscript. Within the domain of a given editor, documents exist in the private form known to that editor. Examples include the Xerox 860 form, the Xerox Star form, or any of the innumerable forms generated by various word processing programs operating in a personal computer. Each of these private forms deals differently with the expression of a document's content and structure, and generally cannot be used by any other editor.

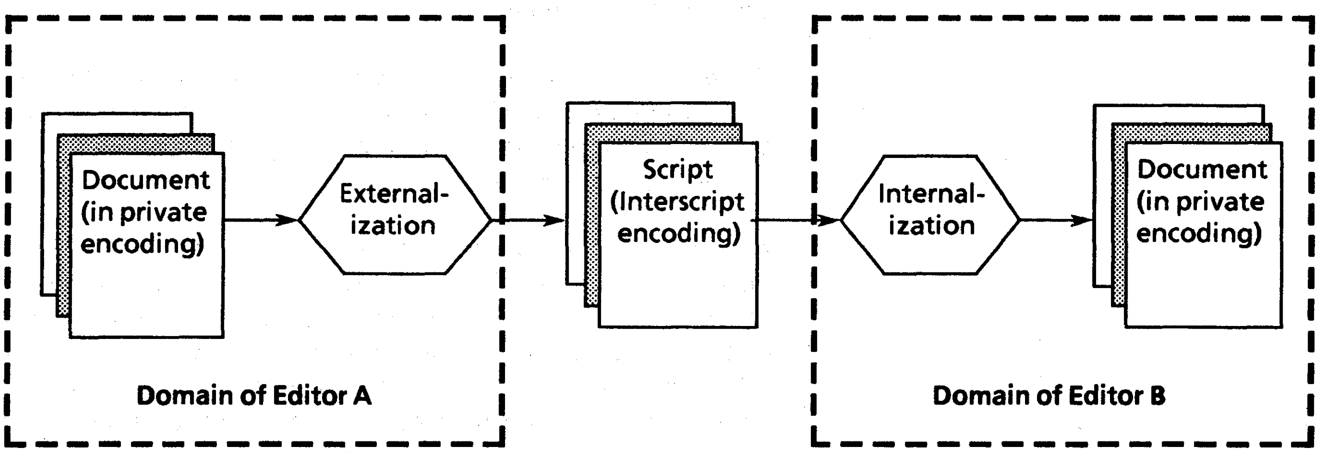


Figure 8-1 **Document interchange using Interscript**

The conversion of a document from its private form to a script is called "externalization." Similarly, the conversion of a script to a private form is called "internalization." These two processes are accomplished by computer programs typically (but not necessarily) associated with the domain of the corresponding editor.

One immediate advantage of this is economy of conversion. If a system has 15 incompatible editors and complete compatibility is desired, only 15 pairs (internalization plus externalization) of conversion routines must be written, rather than the

105 pairs that would be required for the everything-to-everything conversion.

Another advantage is that by careful design, no restrictions are placed on document complexity because of interchange encoding. Problems may exist when a document produced by a feature-rich editor is edited by a basic editor. This, however, is not a problem caused by the interchange encoding.

The design of Interscript is particularly appropriate for a globally-distributed document management system in which documents will exist for years and where continuing changes —upgrades, extensions, new applications—are the norm.

Interscript is a kind of computer language for representing the content part, logical structure, and layout structure of documents (see Fig. 8-2). All three are important.

Content part: What you see when you look at a document.

Logical structure: The way a document is organized, typically but not always in a hierarchical structure (paragraphs subordinate to sections which are subordinate to chapters).

Layout structure: The way a document is to be rendered (headings and footings on every page, multi-column dimensions, margin settings, and the like).

Interscript as a document interchange standard

Some Interscript qualities that make it a general-purpose document interchange standard include:

Encoding efficiency A script encoded according to the provisions of Interscript does not take up a particularly large space. Such a script is well-suited for transmission and intermediate storage.

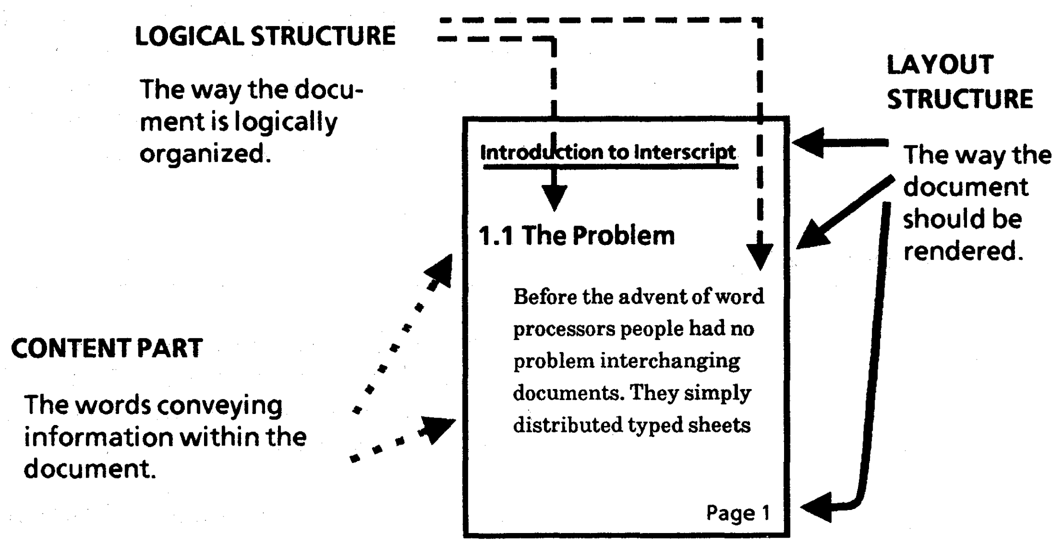


Figure 8-2 **A document's constituent parts**

Open-ended representation

The power of the document interchange service is only as great as the comprehensiveness of the form into which documents are forced for interchange purposes. Interscript, therefore, is comprehensive relative to forms of data, types of edit functions, etc. But it is also extensible, so that as new constructs come into widespread use, it will be possible to incorporate them into the interchange coding without perturbing existing encoding provisions. This means that as Interscript is extended, old editors will still be able to work on old material (internalized from the Interscript encoding) without modification.

Document structure

A careful association is maintained between the content of a document and the form attributes applicable to various parts of the content. If original documents are hierarchical (e.g., a book with sections, chapters, paragraphs, etc.), that relationship is maintained; but where a hierarchical relationship does not originally exist, none is imposed. Some documents contain indirect (non-hierarchical) structural relationships between separate parts, such as the practice of printing informal comments in a formal specification in some specially-reserved type face, or the practice of interlocking certain graphical elements according to their geometries. These indirect structural relationships are preserved.

Transcription fidelity

Interscript-encoded documents are capable of generation and consumption in such a way that information is not changed. A document can be externalized into Interscript, then internalized back into the original private form, without incurring any change.

Interscript also makes it possible for the simple parts of complex documents to be understood by simple editors. This means many simple workstations may be used for routine editing chores. Although a basic editor might not be able to handle all the constructs of an internalized document, the user can do whatever the editor is capable of doing.

With Interscript's design, the internalization/externalization processes can be structured to allow the characteristics of an internalized document (to which the editor has made no changes) to flow through the editing process and remain intact when the document is once more externalized.

Interscript base language

The Interscript base language defines the representation of documents in a byte stream. This digital representation encoded using the Interscript base language is called a script. A script is an encoding of the document, not the document itself.

Interscript allows the definition of symbols with which some formal meaning is associated. These symbols are then interpreted by Interscript systems according to their precise meaning. The Interscript base language also has these features which together make it unique:

- Complex data structures may be represented in the form of a data stream;
- The contents of a document may be interpreted according to meaningful symbols recognized in the scripts;
- New abstractions that can be associated with symbols may be formally defined. One does not need a meta language to describe the constructs used in scripts; the Interscript base language is sufficient.

Simplicity and extensibility

Simplicity is achieved by a single language. Interscript has another major advantage: one can *insert the definition of new constructs* into a script's data stream using the Interscript base language. This makes it possible for a system to define its own abstractions in terms of the Interscript language, then communicate those definitions together with the scripts which use them. Other systems can then understand these non-standard constructs.

Interscript provides a comprehensive and simple binding mechanism. A binding associates a name with some value; the value is then accessed through the name. This very general operation may also be used to achieve compactness. A form might contain, for example, the same item repeated 25 times. A compact script would bind it to a name and repeat that name 25 times.

The part of the Interscript standard concerning the base language is called Layer 1 of the standard. The entire grammar for the base language may be described in Backus-Naur Form in just 22 rules (see Fig. 8-3).

In addition to the base language, the Interscript standard also provides a set of abstractions that are expected to be widely used. These abstractions are described using the Interscript base language. Since those abstractions have their own significance and are build on the top of Layer 1, this part of the Standard is called Layer 2. Convenient abstractions such as appendix, glossary, codicil, tables, spreadsheets, graphics, etc. can be designed for a particular application using the Interscript base language. The Interscript structure also allows other standard forms to be included within it, such as the Raster Encoding Standard for pixel arrays, and the GKS standard for graphics.

The advent of Interscript and the internalization/externalization processes for editors will launch a new era for modern office systems. A widespread exchange of work products will facilitate new levels of productivity. This is another example of integration in the XNS architecture leading to improved forms of document management.

Rule 1 identifier :: = letter idTail

Rule 2 idTail :: = *empty* | idTail (letter | digit)

Rule 3 script :: = "INTERSCRIPT/1.0" node "ENDSCRIPT"

Rule 4 node :: = "{" items "}"

Rule 5 items :: = *empty* | items item

Rule 6 item :: = tag | localBinding | staticBinding | expression | reference | formula | scope

Rule 7 basicObject :: = INTEGER | "<" STRING ">" | ATOM

Rule 8 tag :: = identifier"$"

Rule 9 localBinding :: = name " = " expression

Rule 10 name :: = identifier | qualifiedName

Rule 11 qualifiedName :: = identifier "." identifier

Rule 12 expression :: = term | operation term | expression operation term

Rule 13 operation :: = "+" | "-" | "*" | "/" | "!" | "<" | ">" | "< =" | "> =" |
"equal" | "not" | "or" | "and"

Rule 14 term :: = basicObject | reference | node | "(" expression ")"

Rule 15 reference :: = (path | reference) " |"

Rule 16 path :: = name | path ":" name

Rule 17 staticBinding :: = name " = = " (expression | selection | formula | quotedExpression)

Rule 18 selection :: = "OneOf" " (" alternatives ")"

Rule 19 alternatives :: = *empty* | alternatives identifier

Rule 20 formula :: = expression"%"

Rule 21 scope :: = "[" items "]"

Rule 22 quotedExpression :: = """ expression """

Figure 8-3 **Interscript base language grammar**

Document Interchange Service

The XNS architecture includes Document Interchange Service that can convert documents to and from different standard formats. These services are available as part of Gateway Services (such as Interactive Terminal Service and the 850/860 Gateway Service) and in workstations.

The Xerox Professional Workstations provide software to convert to/from the format of the older Xerox 860 Information Processor. They can also convert a document in their format to and from the DIF format used by many vendors of word processing equipment. Although this helps alleviate some document interchange problems, it is not a complete solution.

9. MAILING

One of the most useful aspects of a distributed network system is the ability of users to send and receive electronic mail. Electronic mail may be defined as a non-interactive document or message communication between people that is transported electronically, not physically. The usefulness of an electronic mail service depends largely on its wide availability and on its flexibility.

Computer based electronic mail systems offer many benefits that are not available in other forms of communications such as the telephone, telex, and conventional hard copy mail. Some of these features that increase productivity and effectiveness are:

Speed and flexibility

Mail is delivered almost instantaneously whether they are going to a user in the next office or in another city. This eliminates the delay present in conventional mail.

The users also have the advantage of being able to send and receive communication at a time and place of their own choosing. No more telephone interruptions or "telephone tag."

Multiple distribution

Mail may be sent to distribution lists with no extra effort. It is easy to be effective in communicating to a single person, a group, a department, or a whole organization.

Precision and productivity

The written word itself leads to precision. The document is a record of the communication and may be acted upon, increasing the effectiveness of individuals and groups. In addition, powerful electronic mail systems make it very easy to respond to mail, forward it, use parts of it or all of it in some other communication, and file it electronically. It saves time for an individual worker and makes it easy to build upon the work of others.

Standards and interconnection

There are a large number of electronic mail networks in existence today, most of which are incompatible with each other and offer very limited features. To make electronic mail as universal as the telephone or the postal service, it is essential that there be standards for electronic addressing, delivery, verification, and other mail functions, as well as for mail and message formats. The great variety of electronic mail systems also need to be interconnected into a unified network. For electronic mail, internetworking is vital because of the *critical mass effect*; that is, how effective an electronic mail system is depends primarily on how many people it can reach. Since many potential correspondents will in practice be users of different mail systems, interconnection is vital for maximum utility.

Xerox has been a leading proponent of such industry standardization and had a leading part in the development of the CCITT X.400 series of standards for message handling systems. The Xerox mail protocol and mail format standards are aligned with the CCITT standards.

Mailing standards

The Xerox mailing standards consist of two protocols and a format standard. The Mail Transport Protocol and the Inbasket Protocols address the functions of sending and receiving mail, and the Mail Format Standard defines the format of messages transported using the two protocols. Together these three standards specify a carefully layered mailing architecture that promotes compatibility while also allowing flexibility through transparent multi-level encapsulation.

Mailing protocol model

The relationship of the two mailing protocols can best be explained in terms of the CCITT X.400 architectural model which defines two layers: Message Transport and User Agent, as shown in Fig. 9-1. The model envisions an originator and a recipient user agent, ordinarily a human/workstation client although entirely automatic processes might also originate or receive mail.

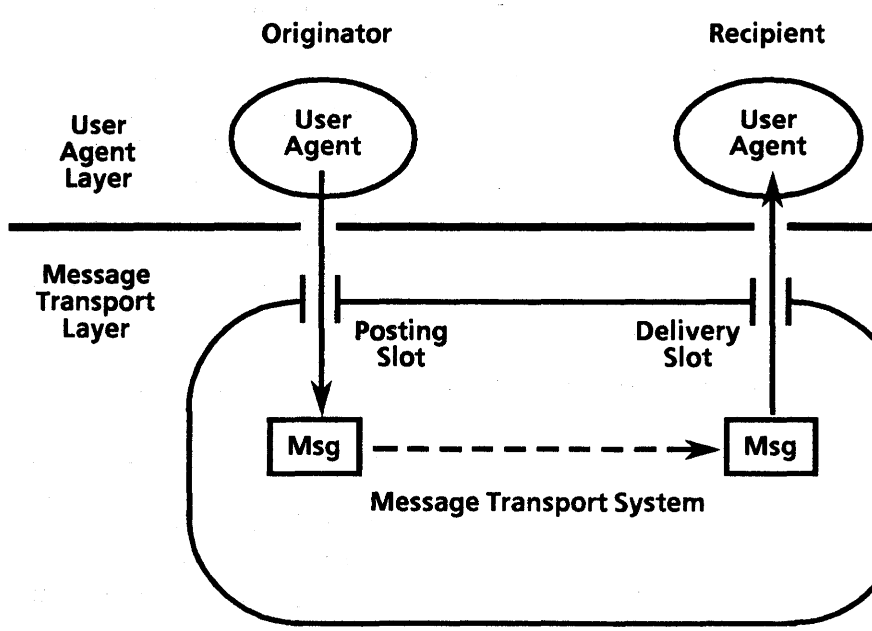


Figure 9-1 **CCITT X.400 architecture model**

Mail Transport Protocol and Inbasket Protocol

The XNS Mail Transport Protocol corresponds to the boundary between the Message Transport layer and the User Agent layer, and provides operations for sending and receiving mail using the posting and delivery slots (see Fig. 9-1). Accessing the XNS Mail Service according to this protocol thus equates the physical boundary between the workstation and the server with the architectural boundary between the two layers. However, there are pragmatic reasons for departing slightly from this model. A typical user agent (client) will transfer incoming mail from the delivery slot to an "inbasket" container (mail file) for perusal by the recipient. Using the Mail Transport Protocol for delivery implies that the mail is transferred to an inbasket container residing on the client machine (e.g., workstation). Since users may wish to gain access to their mail from any of several workstations, it is preferable that the inbasket resides on the server, making it globally accessible. To this end, the Inbasket Protocol is defined, which corresponds architecturally to an internal interface of the user agent layer.

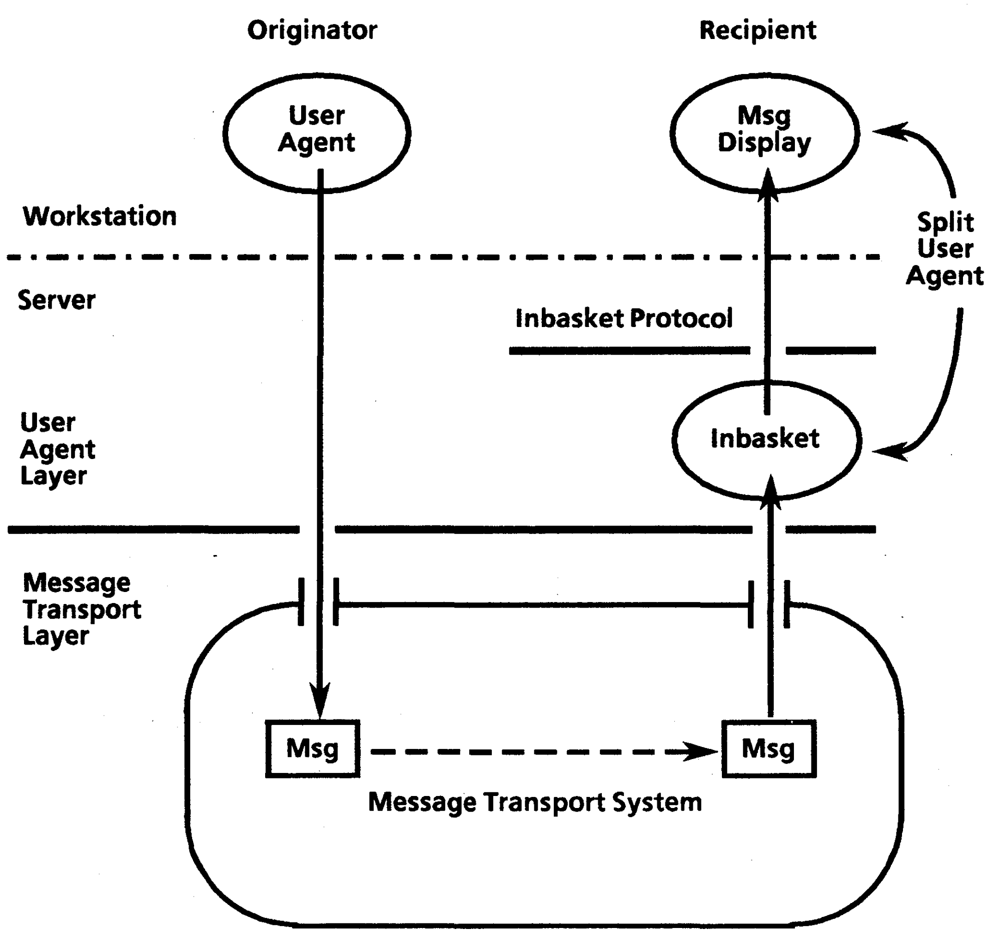


Figure 9-2 **The Inbasket Protocol provides an internal interface for user convenience**

Workstation/service interface

The interface between the workstation and the service thus consists of both protocols: The Mail Transport Protocol is used for posting messages, while the Inbasket Protocol is used for receiving them.

To send a properly formatted message, the originator invokes the Mail Transport Protocol, handing the message through the "posting slot." The Mail Transport Protocol is responsible for calling the appropriate Courier procedures that will deliver the mail to the intended recipient(s) through "delivery slots." The Mail Transport Protocol then delivers the mail to a holding facility managed by the Inbasket Protocol. This protocol acts at the request of the receiving user by fetching messages that have arrived at the holding area since the last access.

Together, the two mail protocols provide a transport medium for messages (including all variety of electronic information) that is totally transparent to the content and format of the messages. Specifically, a message at the Mail Transport level is defined to consist of two parts: envelope and content, as shown in Fig. 9-3.

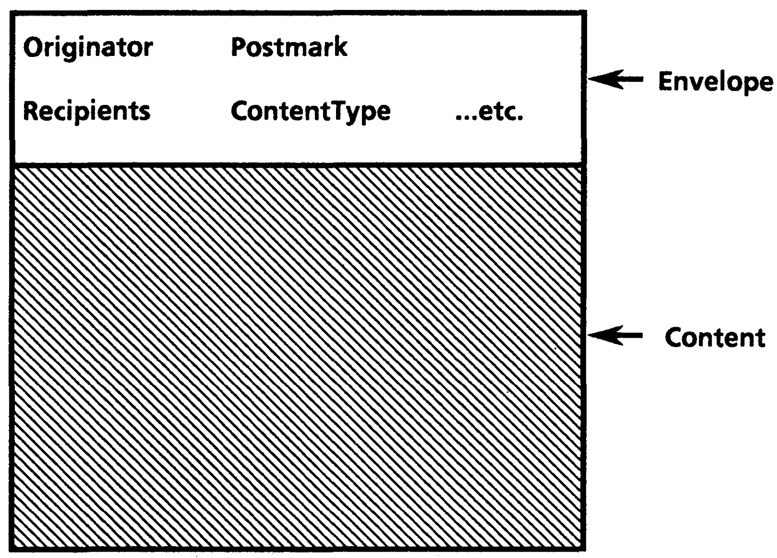


Figure 9-3 **Envelope and content of a mail message**

Content and envelope

The crosshatching in Fig. 9-3 indicates that the Mail Transport level is completely transparent with respect to the messag content. The envelope contains information of two kinds:

1. Information that is related to the functioning of the Mail Transport level, including information passed from the user agent to the Mail Transport system (e.g., recipients) and information passed from the Mail Transport system to the user agent (e.g., postmark).

2. Information that is passed from the originator's user agent to the recipient's user agent and is required on all messages (e.g., ContentType).

Mail format standard

While the transparency of the Mail Transport system provides layering and flexibility, most usage of Mail Transport will adhere to a single format for the content. This format is defined in the Xerox Mail Format standard, which is functionally aligned with the CCITT X.420 P2 specification for interpersonal messages. The presence of such a message is signaled by the appearance of the corresponding ContentType on the envelope. Messages of this type are thus encapsulated within the outer level defined by Mail Transport and provide, in turn, a second level of encapsulation by defining a two part structure consisting of a message *heading* and a message *body*, as shown in Fig. 9-4.

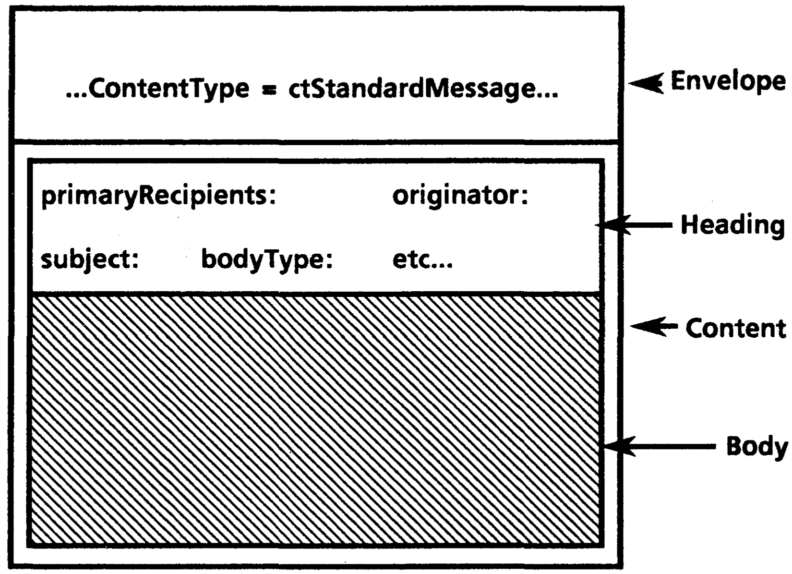


Figure 9-4 **Heading and body of a mail message**

Message body

The crosshatching in Fig. 9-4 indicates that the Mail Format standard is transparent with respect to the format of the encapsulated message body. Examples of message body types would include document formats for various document preparation systems (e.g., Xerox 8010, 860, etc.); these correspond to subtypes of the Nationally Defined body type of CCITT X.420 model. In all cases, the heading information (originator, subject, etc.) is represented in a uniform way, independent of the format of the body. At the innermost level, the mail message body must be interpreted by appropriate format-specific software, selected according to the type indicated in the bodyType field of the heading.

Mail Service

The XNS Mail Service allows users to send and receive electronic mail varying from brief messages to long documents in a variety of formats. Mail is delivered almost instantaneously whether it is going to a user in the next office or in another city. The Mail Service uses the Mail Transport Protocol, the Inbasket Protocol, and the Mail Format standard in its operation.

Accessibility

The Mail Service is directly accessible to all XNS workstations and to many non-networked, non-XNS devices that can dial up the appropriate gateway service.

Mail originating from geographically dispersed networks can also be exchanged across the internet via the External Mail Gateway or the Internetwork Routing Service.

Flexibility in content

Unlike most electronic mail systems, XNS Mail Service supports not only messages but full attachments which can include typeset text, graphics, record files, or any other electronic information. Information originating from a workstation can be sent through electronic mail and be received at any other workstation with all formatting retained. For example, when a document containing tables, various font sizes and faces, and illustrations is mailed, it can be viewed, printed, and filed with all the formatting information retained.

Because the Mail Service can handle all types of information, it provides a convenient means for people to exchange documents as well as data.

Distributed mail system

The Mail Service is a distributed system that allows multiple Mail Services on an internet to cooperate actively to form a unified mail system. It does not restrict the number of Mail Services allowed on an internet or the number of individuals who can communicate, but works as a large cooperative mail system.

Multiple Mail Services act much like multiple post offices within the U.S. Postal Service. Each Mail Service is capable of accepting mail, holding mail for pickup by nearby recipients, and forwarding mail to the recipient's Mail Service.

A distributed mail system has advantages over an independent mail system because of its reliability and potential for growth. It provides constant communication availability because mail can be sent using any Mail Service if a local Mail Service is not operating. A distributed system also allows for smooth expansion from a small configuration to a large system with many users, since additional Mail Services can be added as the number of users grows.

Distribution lists

The recipient list determines who will receive the message. In the XNS Mail Service this list can contain the names of individuals and/or the names of distribution lists. Generally, mail is addressed and sent to an individual, but in many cases documents or their copies are also sent to many individuals. This can be accomplished either by entering the names of all the recipients, or by addressing mail to a previously created distribution list.

A distribution list is composed of the names of a group of individuals who are registered in the Clearinghouse and are designated as a user group. User groups are generally created by a system administrator, but users can be granted administrator privileges when a group is created, allowing them to add and remove individuals from the group. A distribution list itself may contain other distribution lists.

User group distribution lists can also be used to control access, such as access to file drawers. When the name of a user group is entered as a recipient, it becomes a distribution list and all the individuals associated with the list receive the message. All registered users can use any of the established distribution lists when sending mail. This feature saves time and assures consistency and accuracy when mailing to a group.

When a distribution list is used, the Mail Service generates a list of individuals by interacting with the Clearinghouse to determine the users on the list. If several lists are used and there are duplicate names, the Mail Service sends the message only once to each intended recipient. After the recipients are determined, the Mail Service begins distribution of the message. Local recipients with mailboxes located on the distributing Mail Service receive a copy of the message in their mailboxes. Remote recipients with mailboxes on another Mail Service, require forwarding of the message to their own local Mail Service, which places the mail in their mailbox.

Mailboxes and user names

Separate mailboxes are maintained for every authorized user on a Mail Service. By holding mail in a mailbox, the Mail Service allows authorized users to read their mail from any workstation or terminal with access to the Mail Service through the Interactive Terminal Service. Once the message has been copied to a recipient's workstation disk, it can be deleted from the Mail Service mailbox.

Mailboxes are generally identified by the name of the user and the name of the Mail Service where the user is registered. If no Mail Service name is given, the default domain is assumed. XNS permits great flexibility in the choice of user names (all characters except ":" and the wildcard character "*" are allowed). It permits users to choose their own full legal name (e.g., John Q. Public) or any name of their choice. XNS also allows use of aliases and assists clients to find valid names when they have only partial information.

External Mail Gateway and Teletex Gateway Service

The External Mail Gateway feature of the Mail Service allows multiple Mail Services on different internets to exchange mail, but not share other services. This feature is useful when limited access is desired (e.g, a company wishes to send orders to a supplier via electronic mail, but neither party wishes to allow the other any access to its File Service, which would contain private information).

Mail can be sent from one internet Mail Service to another over autodial telephone connections established by the sending gateway. When a user sends a message, it is placed in a special queue which the Mail Gateway software polls periodically. When a message is found, an attempt is made to establish a telephone connection with the External Mail Gateway in which the recipients are registered. These connections are attempted according to an established schedule and are always made by the sending Mail Gateway. Once a connection is made, the message is transmitted and placed in the recipient's mailbox.

Another XNS facility is the Teletex Gateway Service which interconnects the Mail Service to the Teletex services offered by many common carriers (PTT's) in Europe. This allows XNS workstations and other Mail Service clients to send or receive documents from Teletex and Telex terminals.

Interconnection with other non-XNS private and publi electronic mail services supporting the CCITT X.400 standar protocol will also be possible in future.

A crucial aspect of integrated office systems is producing hard copy versions of documents. With the proliferation of workstations, terminals and personal computers, and with the development of sophisticated document creation systems, the need for printed documents has grown enormously. A hard-copy is needed to read, to proof, to distribute, and to combine with other documents.

For historical reasons, printing has a special place in the Xerox approach to integrated office systems. Xerox is the world leader in plain paper photocopying, having been largely responsible for creating that industry in the 1960's with the development of the 914 copier. Since then, Xerox has introduced a variety of copiers and duplicators (high-speed, production-oriented copying systems), each outstanding in copy quality, productivity, reliability, and user convenience.

During the 1970's, Xerox pioneered the electronic printing revolution with the introduction of the 1200, 9700, 8700, 5700 and 2700 electronic printers. Electronic printing, which is also called laser printing, gives a user the capability to produce high-quality print images quickly and at comparatively low cost.

Electronic printing is the preferred approach in XNS for producing hard-copy because of the enormous flexibility it provides in printing all variety of complex images, including typeset quality documents, line graphics, and even photographs that have been converted into electronic form by a scanner. It also gives a user the ability to print a collated set of documents printed on one or both sides of the paper and stapled.

Electronic printing

For years, word processing equipment produced documents in hard copy form using inexpensive character impact printers. But now that images and sophisticated graphics are becoming a part of written material, non-impact electronic laser printers are becoming essential.

Fig. 10-1 shows how electronic printing works. Printing is accomplished by a print server which consists of electronic laser

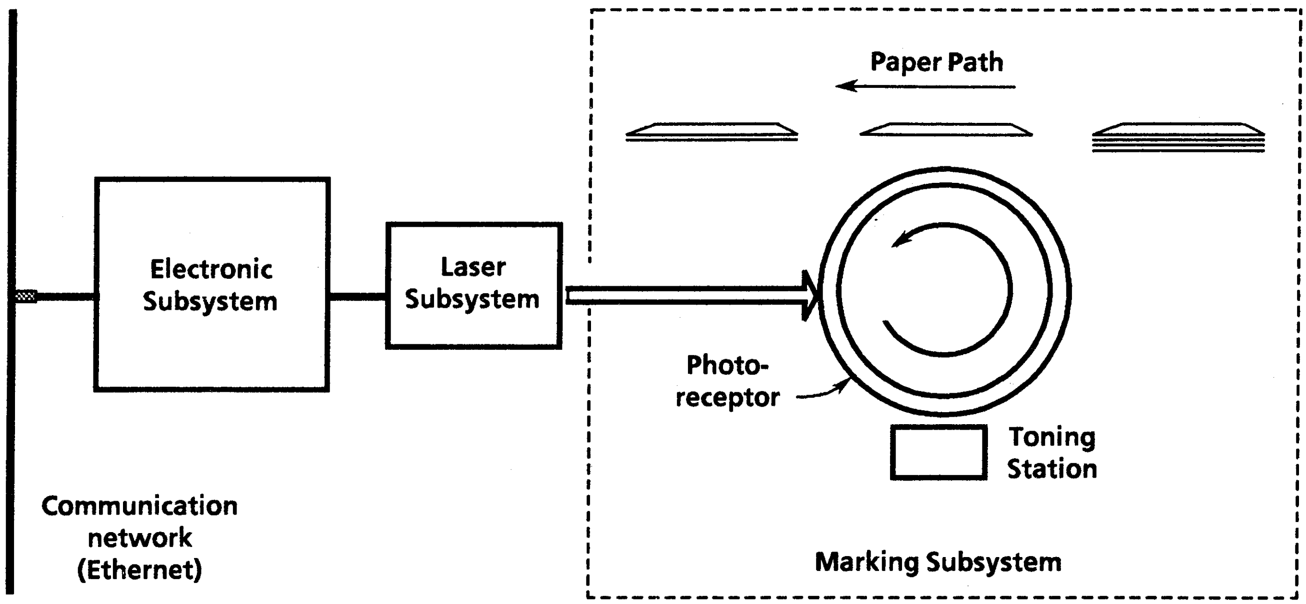


Figure 10-1 **Electronic printing**

and "marking" subsystems which work together to place an image on paper.

In XNS, the electronic subsystem usually receives information over the network (Ethernet or the larger internet). This information consists of data to be printed and instructions on how it is to be printed. The electronic subsystem converts the input information to primitive binary (dot-pattern) form, and drives a laser subsystem which produces a tiny beam of laser light. This light strikes a special surface known as a "photoreceptor," designed to retain the delivered image in the form of a distributed electrostatic charge. The latent image on the photoreceptor passes over a "toning station" which applies a thin layer of black toner to the photoreceptor. The toner sticks to the receptor in locations determined by the electostatic charge. As the photoreceptor moves, it comes in contact with a piece of paper placed in position by paper-handling mechanisms; the tone image is thus transferred to paper.

As the photoreceptor moves, the laser beam exposes it by traveling back and forth across it. This is similar to the way in which an image is constructed on a television screen. Within each trace across the photoreceptor, the laser beam is switched on and off according to the binary values that make up the image. Between the on-off pattern within a line and the successive lines along the photoreceptor, an information density of 90,000 pixels (picture elements or "dots" or "spots") per square inch or 300 spots per inch (spi) is commonly used. In advanced print servers, this denisty may be as high as 360,000 pixels per squre inch (600 spi). By contrast, simple dot matrix printers and CRT displays employ a print density of about 2500 to 10,000 pixels per square inch (50-100 spi). Electronic printing, therefore, has the ability to create hardcopy images approaching, and in some cases surpassing, the image quality of conventional offset printing. Because of the enormous flexibility of the "bit map" printing technique, nearly any image that can be imagined may be printed.

Electronic printing advantages

The following are just some of the specific advantages of electronic printing:

—Document quality equivalent to typeset documents with a wide variety of type fonts, many of which duplicate the "typeset" appearance of formally published material

—Wide variety of type sizes and variable text orientation

—Printing graphic elements, such as rules, shadings, line drawings and other generated graphics

—Two-sided and 2-up (or N-up) printing with higher information density resulting in smaller documents and paper-saving

—Merging of printable material from several sources to create complex documents and the ability to customize documents

—High throughput rate from several pages per minute to two pages per second

Many of these advantages are made possible or enhanced by the use of Interpress, the Xerox electronic printing standard. While Xerox electronic printers support many other industry formats (such as IBM computer output formats), the documents sent to an XNS Print Service should be in the Interpress format.

Interpress

Interpress is the Xerox standard for documents in final form to be rendered into "marks on paper" by an electronic printer. It is an outgrowth of many years of research into printing technology at the Xerox Palo Alto Research Center. Interpress is a highly sophisticated approach to printing which reflects Xerox' experience with distributed systems and a conviction that a single, unified document description scheme is required for printing in a network environment.

Interpress model

Interpress *standardizes the interface between the creator of a document and a printer of that document*. In the absence of such a standard, every workstation would be required to contain software for driving a wide range of printers–any printer that a user wishes to attach to the workstation, either directly or indirectly through a network. By standardizing the representation of documents to be printed, each workstation needs only a single interface for printing. This ensures that documents created at different places and on different output devices would be printed clearly, accurately, and reliably throughout the network.

To meet the requirement of a general-purpose standard, Interpress was designed as a language, fitting into a model of the printing process in which the printer is directed to place specific marks on the page at specific locations. In other words, an Interpress master represents a series of instructions to the printer. The printer follows these instructions just as a computer follows the instructions found in a program. This technique makes possible the printing of substantially more complex documents than would be possible with a static specification of a format, as often done with simple character printers.

Interpress is a printable representation (as distinguished from Interscript which is an editable representation). The assumption in the Interpress model is that the document is only converted to Interpress when it is ready to be printed. After conversion to Interpress, only minor adjustments may be performed.

The Interpress model assumes that the creator (usually a user workstation or processor) sends an Interpress master (representation) to a printer, usually over a network, using the Printing Protocol as shown in Fig. 10-2. It is not necessary for the workstation and printer to be on the same local area network. Interpress masters may also be sent to any file server (using the Filing Protocol) for archiving or demand printing applications. Interpress may also be used in conjunction with other network architectures and media such as magnetic tapes.

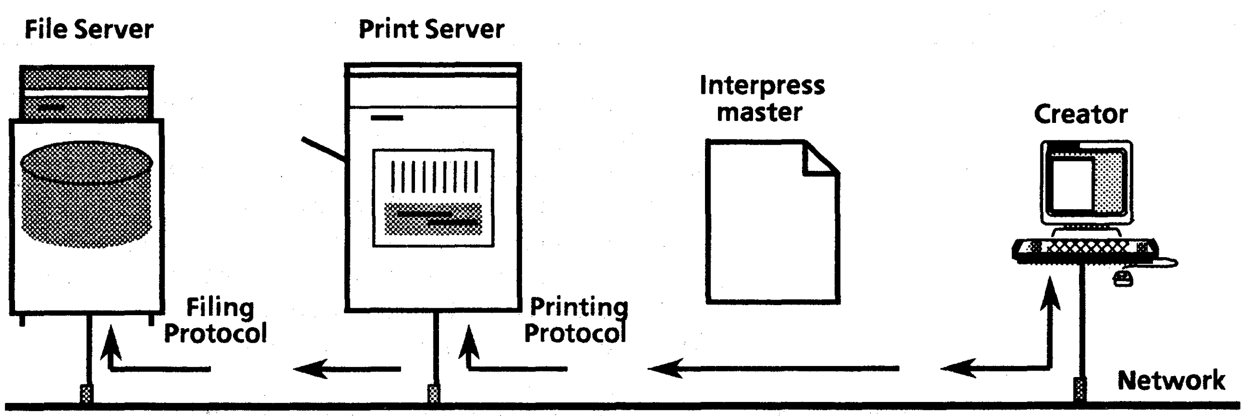


Figure 10-2 **Interpress used for printing or archiving**

Interpress as an industry standard

Interpress has unique capabilities and features and a generalized performance-oriented approach that makes it eminently suitable to be the industry standard for electronic printing. Some of Interpress' advantages are:

Device independence

The standard is independent of a workstation's or printer's specific characteristics, including resolution (the number of pixels in a unit of area), the coordinate system used to locate information on pages in the printer, paper size, the availability (or absence) of specific fonts, and the presence or absence of special characteristics such as the ability to print two-sided pages, color, and scanned images. This allows Interpress to be used for printing on all variety of electronic printers facsimile devices, phototypesetters, plotters and other output devices.

Functional richness

Interpress was designed to take full advantage of the capabilities of today's laser printing technology for high-resolution text and graphics. It is a language for describing and manipulating the pages to be printed. It has a variety of commands for describing text, graphics and pictures; and commands for creating various shapes, rotating them, and scaling them.

Interpress is functionally rich enough to handle a wide variety of applications, including computer printing, word processing, graphical and image printing, publishing, and engineering drawings.

Although Interpress is designed principally for graphics and computer-aided document preparation, it is equally useful for specifying such diverse images as those for 35-millimeter slides, integrated-circuit masks, television stills, or frames in an interactive graphic help system. Almost any two-dimensional image can be specified by Interpress.

Printing control

Interpress contains an extensive set of Printing Instructions which enable the user to control the actions of a printer (e.g., to invoke two-sided printing or special finishing such as stapling). They also provide information necessary for the effective use of a printer within a multi-user environment (e.g., who printed the document, what its name is, whom to charge for the printing, the provision of passwords to control authorization of the final printing, etc.). Printing Instructions also enable the declaration of resources that the document will require (e.g., the files, the fonts, and font sizes). Not only do Printing Instructions enable the control of the printing environment, they also enable an up-front determination of the ability of a given printer to print a document and/or enable it to gather the resources it needs to do so in the most efficient fashion.

Page independent structure

An Interpress document possesses a well-defined structure. Among the attributes of this structure is that it guarantees page independence. Page independence means that the language description of each page is totally independent of that of any other page. Ensuring page independence enables the decomposition and printing of documents in arbitrary page order. (Many printers find it desirable to print last page first. Two-sided printing may require unusual page printing sequences.) Page independence enables the creation of utility routines to manipulate Interpress documents to perform such tasks as creating a new Interpress master from pieces of existing masters, or creating two-up, head-to-toe, or signature masters used in publishing environments. Because of page independence, these operations can be executed easily.

Performance-oriented standard

Interpress enables high-performance printing for simple as well as complex pages, and works economically with both high-speed and low-speed printers. For example, the Xerox 9700 Electronic Printing System decomposes a complete Interpress description of a complex page and prints it in less than a second. Interpress is designed for higher performance and efficiency for the one specific purpose of electronic printing. It is not intended to be a general purpose editable format, a composition language, or to be created by people. The Interpress model assumes a user creates Interpress masters via any number of document editing and composition systems, including graphics composition languages. It is a printing standard with emphasis on an organization that presents a clear separation between the processes that belong in the creation domain and processes that belong in the printing domain. This is one reason for its efficient encoding and execution.

Compact encoding

Interpress specifies a compact encoding for the transmission of a document. It includes special encoding notations for compact representations. The encoding techniques substantially reduce the number of bits required to either store or transmit a document. Simple utility routines are also available for the bi-directional translation from this compact format to full English language form ASCII character representations suitable for human usage.

Total environment

Interpress has all of the essential qualities of a stand-alone language. However, Interpress has been designed so it also fits well into an applications and network environments such as the total XNS environment. For example, Interpress contains a sequenceInsertFile function that enables the inclusion of files accessible by the printer. Such files can contain Interpress masters, fragments of Interpress masters, or even printer dependent object code that represent previously decomposed Interpress masters (e.g., forms). The printer can utilize the full XNS file pathname so an Interpress master can "reach out" into the entire XNS internet that is attached to the printer's local area network.

Interpress takes explicit steps to establish a universal name space and to provide for a central registry mechanism for the distribution and control of names in this space. This provides a means for establishing a uniform environment for an extended family of distributed printers.

Experience and Extensibility

The Interpress Standard has evolved during Xerox' many years of experience using it in a distributed network environment. This is a testimonial to both the reliability and the extensibility of the standard. Interpress has proven to be gracefully extensible so that as new printing technology, new applications, and other new requirements emerged, Interpress was extended without requiring the modification of existing encoded "masters."

Interpress will continue to be extended to address new applications. However, to assure consistent performance across the network, a Reference Subset of Interpress is also defined which all XNS printers are expected to implement.

Interpress language

Like all programming languages, Interpress has both syntax and semantics. The semantics of the Interpress language are the rules for how the various operators behave when they are executed by the printer; the syntax of the language is the set of rules for how the calls to those operators are coded in a master. Because Interpress masters are intended to be created and interpreted by programs and not by people, the syntax is designed to make it easy for programs to produce and interpret, without any concern for human readability.

Interpress is a stack-oriented language which uses a postfix notation (like that used in some hand-held calculators) in which the operands are pushed onto a stack. When the operator is executed, the operands are removed from the stack and replaced with a result. For example, the computation, 5*10 + 8, would be represented in postfix as: 5 10 MUL 8 ADD. Some operators, such as imaging operators, cause an important side effect, such as building a page image, instead of returning a result.

Imaging Operators

The software within a printer interprets, or executes, an Interpress master to print a document. During that execution, the printed document is built up one page at a time. The software maintains a page image; the imaging operators make changes to the page image.

A complex image starts with a blank page image and a sequence of simple changes is made to it. These changes might include placing text characters (possibly scaling or rotating them before placement), drawing lines (horizontal or vertical lines or complex polygons), filling closed areas with shading, or placing facsimile pictures, including halftones. The task of maintaining and altering the page image is done by the imager. Thus the functions of the printer software divide neatly into two parts: a base language interpreter and the imager. As a master is executed, calls are made to imaging operators in the imager which build up the page image.

Consider this simple Interpress program:

```
0.07366 0.23876 SETXY
<Print this> SHOW
```

The first line sets up a starting position for the text string; the second line places a representation of the character string "Print this" on the stack and calls the imaging operator SHOW to place the characters in the page image.

Line drawings and font information may similarly be added and more page bodies placed, as shown by the Interpress master in Fig. 10-3. Line 0 and 14 are part of the skeleton containing two page bodies, one for the rectangular box (lines 1-7), and the other for the text string "Print this" (lines 8-13). Line 2 sets an image variable named strokewidth to 1mm (.001 meter). This variable is examined by the MASKVECTOR operator to determine the width of the lines in the rectangular box. Lines 3 to 6 specify the (x, y) coordinate pairs (in meters from the lower left corner of the page) for MASKVECTOR which draws the lines of the rectangular box. Line 7 ends the first page, and Line 8 begins the second page. Lines 9 and 10 establish a character font and set it as the "current font."Line 11 sets the "current position" to x = 2.9 inches, y = 9.4 inches (expressed in meters), and Line 12 shows the ten characters "Print this" in the current font.

This example images two separate pages. If the page boundaries (lines 7 and 8) were eliminated, only a single page would be printed with "Print this" inside the rectangular box as shown on the right in Fig. 10-3.

The 242-page *Introduction to Interpress* (see bibliography) discusses Interpress capabilities in detail. It is recommended for anyone with a serious interest in Interpress. The Interpress standard itself is written for precision and accuracy (with few explanations) and is the authoritative definition of Interpress.

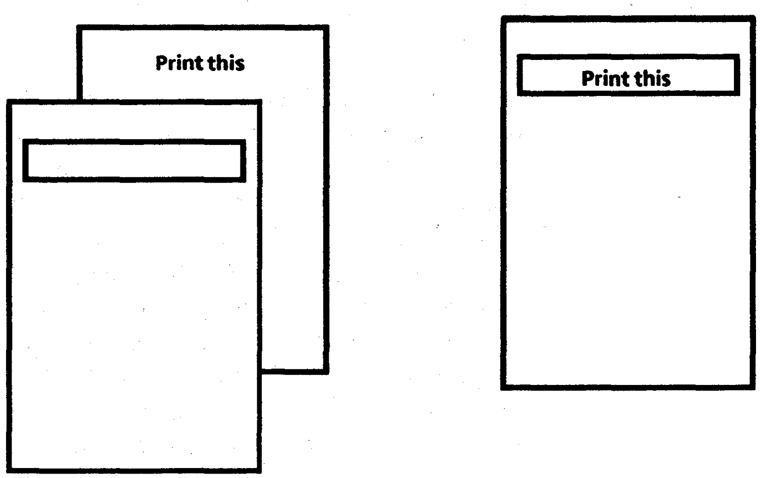
Typographic Printing

Typography is the art of designing and placing letterforms to create a legible and pleasing effect. Interpress is able to describe documents of extremely high resolution and quality, and thus is able to represent documents with fine typography.

The quality of typography depends upon the availability of character sets and fonts. The Interpress standard is used with the Xerox Character Code Standard and the font architecture. Interpress allows fonts and arbitrary character sets to be used

Line Number (ref. only)	Interpress Master	Comments (for reference only)
--0--	BEGIN { }	*part of the "skeleton"*
--1--	{	*beginning of the first page body*
--2--	0.001 15 ISET	*set imager variable 15 (strokeWidth) to 0.001*
--3--	0.0254 0.2286 0.0254 0.254 MASKVECTOR	
--4--	0.1905 0.2286 0.1905 0.254 MASKVECTOR	
--5--	0.0254 0.2286 0.1905 0.2286 MASKVECTOR	
--6--	0.0254 0.254 0.1905 0.254 MASKVECTOR	
--7--	}	*end of the first page body*
--8--	{	*beginning of the second page body*
		line 9 defines font and saves it in frame element 0
--9--	[xerox, xc82-0-0, times] FINDFONT 0.00635 SCALE MODIFYFONT 0 FSET	
--10--	0 SETFONT	*sets the "current font"*
--11--	0.07366 0.23876 SETXY	*sets the "current position"*
--12--	<Print this> SHOW	*place "Print this" at current position in current font*
--13--	}	*end of the second page body*
--14--	END	*end of the master (more of the "skeleton")*

Two page images produced by this Interpress master.



One page image produced by this Interpress master without lines 7 and 8.

Figure 10-3 **Example of an Interpress master** and intermixed freely in a master. There is no standard "Interpress Character Set." Interpress does not recognize ASCII formatting characters such as carriage return and tab because there is no accurate way to interpret what these characters should do. Formatting operations are achieved by Interpress positioning commands such as SETXY (see Fig. 10-3).

Interpress also does not generate ligatures automatically. Ligatures must have a separate character code (provided by the Xerox character code standard) and representation. This adheres to an important Interpress design goal: *All decisions about presentation and formatting should be made by the creator, not the printer.*

Interpress provides many facilities to achieve high quality typography including letterform definitions, expressed as character operators (no restrictions are imposed by Interpress); positioning operators used to control the position of the letters; geometric transformations to scale, rotate and translate a letterform so it can appear in arbitrary size, rotation, and position on a page; and additional graphical operators to define underlines, strikethroughs, and the like.

Positioning in Interpress may be either absolute (with respect to the page) or relative (with respect to some other coordinate system such as a box within a page). Interpress also provides a CORRECT operator for spacing adjustment should it be required. Interpress also provides a flexible way to achieve kerning, an alteration of intercharacter spacing of pairs of characters to achieve a better appearance.

Graphics Printing

Interpress has a rich set of imaging operators for producing graphic images. It provides facilities for generating filled geometric shapes, scanned images, and stroke generation operators. The ability to produce images in color or gray is also provided.

The notion of a mask is central to Interpress graphic primitives. A mask can be visualized as a geometric shape (analogous to a silk screen) laid over the page image to determine where to apply colored ink to the image. The mask determines exactly where the image will be changed; the colored ink determines how it will be changed. Interpress provides a rich set of mask operators for specifying masks and an image variable specifying the color of ink. Each mask lays down a single graphic primitive; complex pages are generated by calling mask operators many times, each time adding a stroke, filled object, or scanned image to the page.

A very useful feature in Interpress is the notion of priority to resolve ambiguity about which mask or color will be visible when more than one color is used on a page and masks of two different colors overlap. Some printers may not find it convenient to print objects in the order in which they are created within the master. Interpress contains an imaging parameter, called priorityImportant, that is used to designate when the printing sequence must match the Interpress presentation sequence. When priorityImportant is not true, the printer is free to image objects in a sequence of its own choosing. This can improve printer performance in many cases.

Utility programs

The programmability built into Interpress allows a master to be created that will automatically adapt itself to varying circumstances by testing its environment and creating different images accordingly. But there are times when these manipulations cannot be anticipated or are too complicated to build into the master at creation time. These master manipulations are performed by utility programs. Examples of such manipulations are selecting pages from a master to make a new document merging separately created pieces (such as text, graphics and scanned images) into one master, and preparing 2-up or N-up signatures commonly used in the printing industry. It is the page independence built into Interpress that makes these operations possible. To understand the manipulations performed by a utility program, consider signature preparation. This manipulation can be done by changing the geometric transformation associated with each page in the Interpress master to cause a rotation by 90 degrees and a size reduction by a factor of 0.65 to be applied to the page. The various page definitions are then rearranged and combined so that when the sheets are printed, folded, and stapled, the pages will be in the right sequence. This operation is shown in Fig. 10-4. Note that an edge of each signature page is unused because of the ratio of the reduced page compared to its original size. More complex master manipulations of this kind can be used to assemble images for 16-page signatures.

Manipulating a master with a utility program is not the same as editing the document with a document-preparation system. Master manipulations are not intended to correct typographical errors or to insert new text because these changes require formatting decisions and could lead to repaginating the entire document. That is the job of a text formatter, not of an Interpress utility. Typical master manipulation operations are analogous to cutting and pasting with a reduction/enlargement capability.

Printing Protocol

XNS clients use the Printing Protocol to cause documents to be printed on a Print Service. This protocol assumes that the document is encoded in Interpress. Documents not encoded in

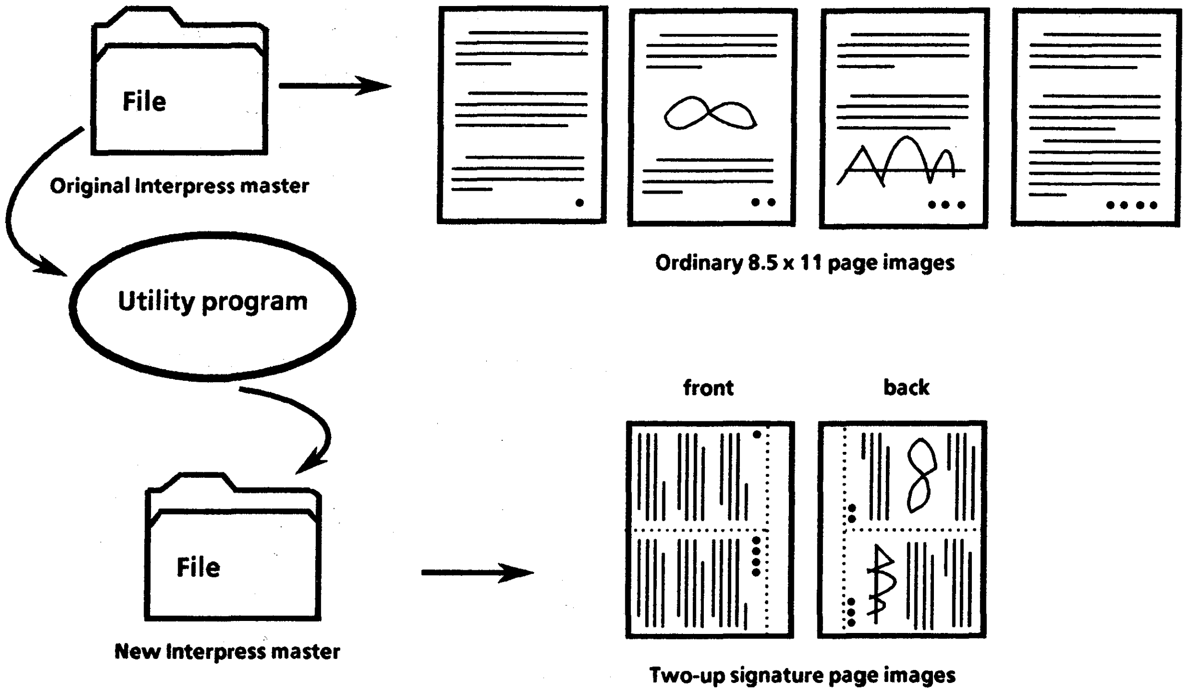


Figure 10-4 **Preparing two-up signature pages**

Interpress can be printed in an XNS system only if special provisions are made by private arrangement between the client and service.

The protocol provides a general printing facility to support a wide variety of applications, but it does not itself support filing, mailing, network administration, and other such functions which are subject of other XNS protocols. (These functions, however, may be supported by the Print Service.)

Printing Protocol model

The Printing Protocol model assumes an abstract printer service which has three distinct processing phases: spooling (queuing), formatting, and marking (printing). In the spooling phase, the Interpress master is queued in a special holding area (which may or may not be in the print server) for subsequent printing. In the formatting phase, the Interpress master is converted to a form suitable for rendering by the specific printer marking engine. In the marking phase, the document is actually printed on the physical medium. The printing model assumes there is a queue (which may have maximum length of zero) associated with each phase holding requests waiting to be processed.

Client request

A client requests service and, if the Print Service is able to grant the request, the client is given a print request identifier. This identifier can be used by the client to track the progress of the request in the three processing phases. (The Print Service supplies the status of the printer.)

Within the print request, the client specifies the document(s) to be printed, as well as a number of options including recipient name, the portion of a document to be printed, the number of copies, 2-sided printing, and stapling. (Some of these options can also be communicated as printing instructions within the Interpress master, as specified by the Interpress standard.)

Priority and security

The Printing Protocol allows documents to be printed with high, medium, and low priority. It also provides a mechanism to assure security, keeping the document from being printed until a user supplies a release key. (This is useful when sensitive documents are sent to a shared Print Service.)

Printer properties and status

The Printing Protocol allows clients to obtain information on the properties and the status of a printer. These are specially useful in a large network when a client wants to use a printer with specific features.

Use of other protocols

The Printing Protocol is Courier-based which means that control information passes from client to Print Service via remote procedure calls. The Interpress master itself is transmitted by means of the Bulk Data Transfer Protocol (which allows the master to be obtained from either the client or third party such as a file server). The client sends the necessary connection parameters for the transfer to the Print Service as an argument of a print request; the printer uses this information to obtain the master. The Authentication Protocol is used to verify user identity. The Time Protocol is used to obtain time and record time during the various printing phases.

Print Service

An XNS Print Service accepts Interpress masters from client programs and prints them on an electronic printer or other output device. Any document in Interpress form may be sent to any Print Service on the internet via the Printing Protocol. The open-ended flexibility of the Printing Protocol and of Interpress makes it possible for a Print Service to support a wide range of output devices including all kinds of electronic printers, facsimile devices, plotters and phototypesetters.

Print Service operation

To use a Print Service, a client (usually a workstation or mainframe program) converts an internal document representation to Interpress and sends this Interpress master to the Print Service via the Printing Protocol. The Interpress master is accompanied by properties and options which specify document name, user name, and creation date. The Print Service interprets the master and directs the printer during the printing process. This process involves the three stages of the Printing Protocol model previously discussed: spooling (queuing), formatting, and marking (printing).

The internal print queue facilitates the movement of documents through the various stages of processing. The queue can hold many documents which are usually processed in the order received. Generally, however, a print job is available in the printer's output tray within minutes after the user initiates the print request.

The Print Service may be shared among the users on the internet. Provisions also exist to restrict access via the Clearinghouse and to ensure security via the Authentication Service. The client, on behalf of a user, can find the status of any Print Service on the internet as well as the status of a particular request.

Electronic printing service

Some examples of XNS electronic printing services are the 8000 NS Print Service (804x Print Servers, NS Electronic Printers, etc.), the 5700 Print Service, and the 8700/9700 Print Services. These support a wide-range of throughput and other capabilities such as one- or two-sided printing, stapling, graphics, scanned images, etc. All of these electronic printing services employ 300 spots-per-inch resolution printers and 300 spots-per-inch fonts. Higher resolution printers may be offered in the future.

The electronic printing services facilitate the printing of all documents including record files, spreadsheets, etc., that are encoded in Interpress. A single printed document can contain several different font styles, in different sizes and intensities, and almost unlimited graphics representations. The services also support the printing of characters ranging from mathematics and science symbols to office symbols and foreign language characters, including Japanese Kanji and other non-Latin alphabets.

A Formatting Print Service (FPS) is also available in XNS to convert complex graphics into compressed scanned images so all standard Interpress documents can be printed on printers which may not have graphic capabilities. Print servers may provide other services as well. For example, the 8700/9700 supports a subset filing service (Printer Subset of Filing) to receive scanned images directly from a graphic input station.

Facsimile printing service

Facsimile Printing Service allows XNS users to share a Telecopier 495-1 (TC495-1) facsimile device. The TC495-1 is a thermal transfer printing device that can serve as a local printer for all workstations, or communicate to distant facsimile terminals for remote printing from a Xerox professional workstation or an IBM Personal Computer.

Facsimile Printing Service supports transmission to all Xerox and non-Xerox facsimile terminals that are compatible with

CCITT Group 1, 2, or 3, allowing documents to be sent anywhere in the world. Documents can be sent to any Facsimile Print Service for transmission by the TC495-1 over the telephone to remote facsimile devices. Many telephone numbers can be supplied at one time, allowing the same document to be sent to multiple locations.

The service also provides automatic redialing when a facsimile device is unavailable. The number of retries is fixed by a system administrator and will not continue indefinitely. As in other Print Services, a user can determine the status of print requests by using the service executive at the server terminal or from a workstation.

Images composed of detailed graphic illustrations, equations, and/or text in a wide variety of fonts can be sent to any Facsimile Print Service. The facsimile device prints images at a resolution of 200 spots-per-inch.

Print Service Integration Standard

In a distributed environment with shared Print Services, it is important to assure that a document printed by different printers will look the same and be of the same consistent high quality. To achieve this, the XNS architecture specifies the Print Service Integration Standard (PSIS). While Interpress standardizes the page description language and methodology, it deliberately leaves certain aspects (such as character codes and font names) unspecified to allow generality. PSIS contains the additional specifications that are required to define the basic elements of the process of achieving full compatibility between document creation systems and Xerox Network System printers.

The PSIS defines the base case to which all XNS Print Services must adhere to assure this compatibility. Many products exceed the PSIS specifications and provide additional capability, but products that conform may not provide less capability. The principal areas addressed by PSIS are:

Interpress level Specification of Interpress as the means for defining the content of a document to be printed on XNS Print Services, and the specification of the Reference Subset of Interpress (defined in the Interpress standard) as a lower bound of Interpress capability.

Character code The establishment of the Xerox Character Code Standard as the means for representing character strings within the XNS Print Services.

Naming syntax The standardization of a naming syntax for a number of critical objects used within Interpress applications and the creation of a set of standard names that adhere to this syntax for these critical objects. These objects include fonts, decompression operators, and colors. A method for allocating and controlling the name space for these entities is established and the use of the name Xerox as the first element in the hierarchical name is specified for the base case standard. The semantics for certain critical hierarchical elements of specific names (e.g., font names) is also established together with a mechanism for their allocation and control.

Font usage

The standardization of a font naming syntax and the establishment of a standard set of fonts that must be resident in every printer.

File usage

The establishment of a standard for file names and the use of the operator sequenceInsertFile. The establishment of a standard for file formats for Interpress files and Image files (defined in the Raster Encoding Standard which includes standardization, naming, and designation of compression algorithms).

Minimal service provisions

The provision of Minimal Clearinghouse and Time Services in very small networks where such services are absent. Also specifies the use of the Printer Subset of the Filing Protocol for the transmission of files to the printer for storage.

Color definitions

The standardization and naming of color definitions including the definition of highlight color.

Printing Protocol usage

The use of the Printing Protocol for the transmission of documents, the standardization of printer error conditions and error messages, and establishment of password conventions.

11. SCANNING

Early document-oriented architectures were considered adequate if they were simply capable of handling text documents. They were considered unusually flexible to the extent they could handle varieties of text forms and could manipulate text in sophisticated ways.

More modern systems recognize that information, particularly that used within documents, can have several forms, of which text is only one. Another important type is "generated graphics" —line drawings, bar charts, etc.

A third important form of information is the scanned image. Such images arise when existing hard copy artifacts (such as photographs, drawings, or signatures) are converted into a digital representation by an appropriate scanning device. This representation consists of the picture elements (called pixels) that make up the image. Since the pixels are an array of binary digits (bits), a scanned image is often called a pixel array or a bit-map. When scanned images are included within documents consisting of text and generated graphics, a composite electronic document of unusually great flexibility results.

Scanned images necessarily require considerable binary information storage. An 8-1/2x11-inch page, scanned at a customary resolution of 300 spots per inch, produces more than one million bytes of information. Compression algorithms can reduce this, but their effectiveness depends on the content of the image (simple text images might compress to as much as 20:1, or about 50,000 bytes, while photographs might compress to only about 2:1 or 500,000 bytes). The XNS architecture defines the method by which the resulting "digitized" image is encoded for subsequent use.

Raster Encoding Standard

The Raster Encoding Standard (RES) is a very general-purpose encoding and image format design that permits an image to be used as a document by itself, or as parts of documents subsequently handled by Interscript and Interpress.

Device-independent representation

RES describes a digital representation for interchange of all raster images (such as those created by scanning devices or by software which outputs rasters). The representation accommodates images of different sizes, resolutions, intensity quantizations, encoding, and compression attributes. A complete description of the image is included, allowing both the printing and editing of the image described.

RES uses Interpress encoding rules and Interpress terminology. This simplifies printer software that handles raster image files specified by Interpress masters. RES can also serve as the description of raster images within Interscript, becoming part of the document description of editable documents. RES image descriptions are like Interpress: they are not intended to be read directly by people, so no attempt has been made to encode the data in human readable form.

Flexibility

In RES, each raster image, whether binary, continuous-tone or color, is described as a single entity (file). A raster scan pattern is assumed (although others could be accommodated by compression schemes) and allowance is made for the possibility of non-rectangular image areas. The representation is flexible enough to permit simple images to be described with little overhead, yet considerable related information can be included when necessary. This flexibility is achieved by specifying encoding rules and results which must occur when the representation is executed according to the rules, rather than specifying a rigid format for the data. Raster image data, which can become very large, is described in compressed form with provision for several standard compression schemes.

The current version of RES includes specifications for binary and gray scale monochrome (B&W) images, as well as highlight or pseudocolor images. Full color images are planned. While not specified yet, color images will probably be compatible with black and white to simplify interchanges between monochrome and color printing and editing systems.

Raster image files

The raster image file contains the information necessary to create the pixel arrays to be edited or printed. The common approach to printing will be by reference to this file, using the Interpress SequenceInsertFile encoding notation. Since RES uses the Interpress encoding rules, it is also possible to include a raster image file in an Interpress master, and execute it. To limit the possible effects of executing a raster image file with its complexity, RES allows only a subset of primitive operators specified by Interpress.

To understand RES encoding, consider the following example of an image scanned on a gray-scale scanner with a resolution of 300 pixels/inch (scanning along the longest dimension). The image is intended to be shown 2 inches across by 3 inches high

Raster Image File	Comments
Interpress/Xerox/2.1/Raster Encoding/1.0 BEGIN	*Header*
254/3000000	*meters/pixel scale in x* *i.e. = ((2.54cm/in)/(300pixels/in))/(100cm/m)*
DUP	*same scale in y*
2 MAKEVEC	*element 1, imageScale*
600	*element 2, xDimension*
900	*element 3, yDimension*
0	*element 4, maskImage; null*
600 900 1 255 0	*data for making pixel array starts with these*
1 SCALE	*the transformation*
[...big vector of integers]	*the compressed data*
Xerox packed 2 MAKEVEC FINDDECOMPRESSOR DO MAKEPIXELARRAY	*element 5, colorImage*
255 0 0 3 MAKEVEC	*vector giving sample meanings*
Xerox GrayLinear 2 MAKEVEC	*color model operator name*
FINDCOLORMODELOPERATOR DO	*element 6, colorOperator*
name <Picture of Lincoln> creationTime <1983 10 12 13:22:15-05:00> 4 MAKEVEC	*element 7, imageProperties*
13086 END	*element 8, signature*

Figure 11-1 **Example of raster image file**

and is, therefore, 600 pixels by 900 pixels. The raster image file for such an image would be encoded as shown in Fig. 11-1.

The file starts with a header (same as Interpress but with an RES version number), and the BEGIN statement. It consists of only eight elements.

The first three elements must be imageScale, xDimension, and yDimension. The next two elements (4 and 5) are the value of the mask or the color (defined in the Interpress Standard) on a two-dimensional grid. The representation of data within RES which results in creation of the pixel arrays is somewhat complex.

Within the raster image file, it is expected that the actual data will usually be contained in a compressed form. The decompression operators (not normally specified within the Interpress standard) are obtained from the environment by use of the Interpress operator FINDDECOMPRESSOR. The result of this operation is to convert the compressed data into an array of samples which are converted to a pixel array by use of the Interpress operator MAKEPIXELARRAY.

Next (element 6) is a color operator obtained from the environment with FINDCOLORMODELOPERATOR. The imageProperties vector (element 7) is normally created by a series of name, value pairs, made into a vector with the Interpress MAKEVEC operator. Finally, the signature identifies the file as obeying the raster encoding standard, and then the END statement.

Image compression

The digital samples that define scanned images can be encoded in several ways. These encodings are able to compress the image data by removing redundant information. Four kinds of image encoding standards are defined in RES:

1. **Packed**. The digital sample values are packed into a sequence of bytes to avoid wasting storage, e.g., eight binary samples are packed into a single byte.

2. **CCITT–4**. The sample values of a binary image are represented according to the compression scheme that is being standardized by the CCITT (Group 4 facsimile apparatus). This representation achieves considerable compression.

3. **Adaptive**. The sample values of a binary image are represented according to a standard developed by Xerox. Compression is achieved by predicting a value based on values of pixels that have already been computed; for example, the predicted value of a pixel may be that of the corresponding pixel on the previous scan line. Up to fifteen different compression techniques are used, each designed to remove redundancy from a certain kind of image: Text characters, line art, and halftones of various screen frequencies. The algorithm adapts to the properties of the image by selecting the technique that will perform best.

4. **Compressed**. Similar to the adaptive technique, but only a single kind of prediction is performed.

This selection of compression methods is based on the following considerations: Compression must not lose information; a wide variety of image content must be accommodated (e.g., halftones, line art, text); scan lines must be of arbitrary length; a range of image resolutions must be supported;

compatibility must be maintained with existing products; and method should allow high-speed decompression and be suitable for VLSI implementation.

RES can be extended to include additional compression techniques (e.g., for continuous-tone images). These new compression methods will be added to the standard, as required.

The following are the guidelines for selecting the compression method for an application: The Packed form is the most general in that it can represent binary and continuous tone images, including images that have more than one sample per pixel (however, it provides no real compression); the CCITT-4 form is an emerging international standard for digital facsimile equipment, and it is recommended for use in all low-speed printers and scanners; the Adaptive form is the standard for all high-speed printers and scanners (generally those with data rates greater than 2 megabits per second); and the Compressed form is used for backward compatibility with several existing products including the Xerox 150 Scanner, Electronic Printer Image Construction (EPIC) host software, and 8700/9700 Electronic Printers with the Graphics Handling Option (GHO).

Image editing

Certain scanning parameters such as those relating to image identification (name, description, time of creation, etc.), scanning characteristics (scanner type, image type, scan transformations, etc.), image processing (image aperture, threshold, masking, etc.), and image statistics are not required for printing but are very useful in image editing. RES defines these parameters as optional and open-ended, and suggests a uniform way of specifying those that may find common usage.

Compatibility

RES provides compatibility with the existing Xerox products that use the IMG Graphic Image Format, used in the 150 scanner and 8700/9700 GHO products. A restricted form of RES encoding makes it possible for existing printers to print the restricted format RES files without modification. RES also allows backward compatibility to IMG so IMG format files can be printed on printers implementing RES. It is for this reason that the IMG format is considered an RES subset.

Scanning services

XNS architecture allows for a variety of models in which scanning devices (or scanners) may be integrated. The two dominant models in XNS (see Fig. 11-2) are:

- The graphic input model in which the scanner is an XNS system element with a user interface of its own.

- The peripheral device model in which the scanner is a peripheral to an XNS workstation (or to a workstation connected to XNS via a gateway service).

A third model is also possible in XNS in which the scanning device is a generalized network service, similar to print and file services. This model is not very practical because the user must take the original documents to the scanner in order for them to be scanned. Considerable interaction may be required between a user and the scanner. It is usually more convenient for a user to have the scanner user interface at the scanner itself (graphic input model) or next to the scanner (peripheral device model). In both dominant models, there is no loss of generality as the scanning service may still be shared among many users, and a user still has access to all of the internet resources for printing, filing, mailing, or communicating the scanned images. RES is used in encoding the scanned image and representing it as a raster image file.

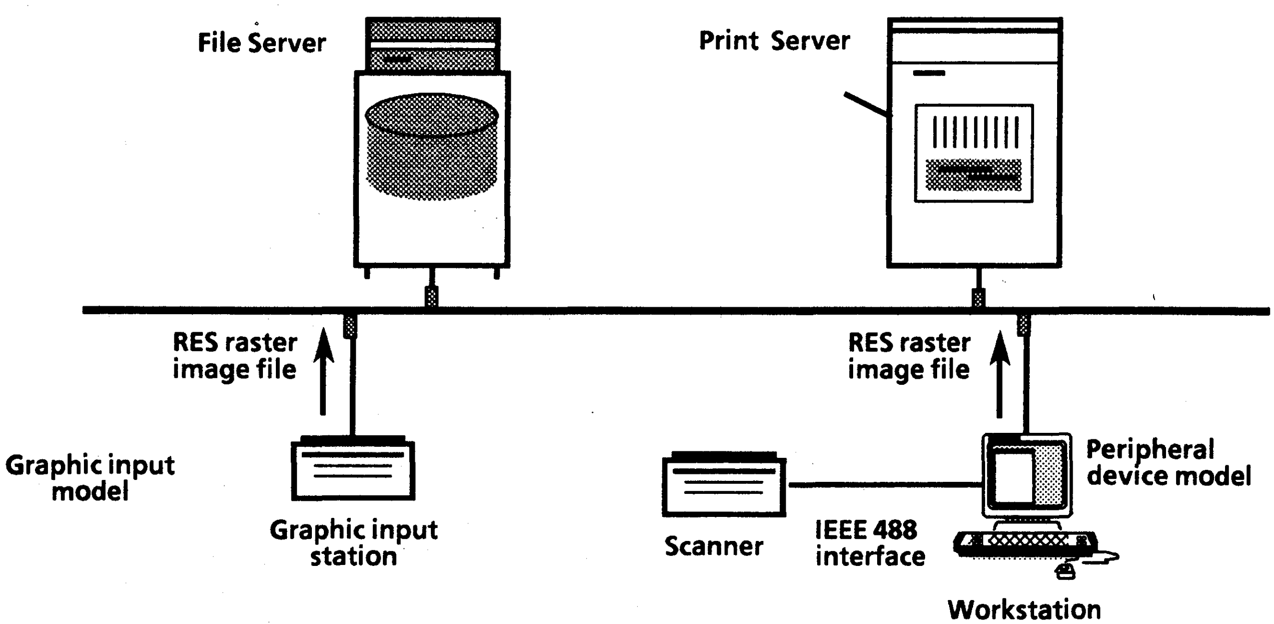


Figure 11-2 **Two models for scanning service**

Graphic input model

This model enables a user to digitize a hardcopy image by scanning it at the scanner. The digitized image (in RES) may be sent to a specified file in a File Service for storage, or to a Print Service for printing (using Printer Subset of the Filing Protocol). A user interface exists at the scanner to allow a user to perform this function, as well as other functions such as cropping or scaling the image. The scanner is an XNS system element which uses XNS protocols to communicate with other devices and services on the internet. The scanned image may be combined with text to form a composite document. The combining can take place at a workstation or at a printer, using the Interpress SequenceInsertFile. The Xerox 150 scanner uses this model in providing scanned image service to XNS users. This model is desirable in production-oriented applications requiring line graphics or photographs.

Peripheral device model

This model is similar to the graphic input model except that it is the workstation and not the scanner which is an XNS system element. The scanner is merely a peripheral to the workstation, usually connected via bus interface such as the IEEE 488. This model has the advantage that the user can do significant editing of the image at the workstation and have a high-level of interaction with the scanner (which is controlled by the workstation) in getting the image digitized to the parameters that best suit an application. This model is best for situations where significant user interaction is desired, such as preparation of high quality original artwork from photographs. The workstation that controls the scanner should have a bit map display and image editing capabilities for this model to be useful.

Intelligent scanning

A variation of this peripheral device model is used by the Kurzweil Intelligent Scanning System, which not only scans a document but converts it into its textual rather than bit-map representation. The Kurzweil System uses the 860 gateway service to connect to the internet (see Fig. 11-3) to store files on a file server or mail them to a user for editing at a workstation. This facility is very useful for all kinds of data and document entry and is a real productivity aid when a large amount of information must be transcribed into text characters rather than image form.

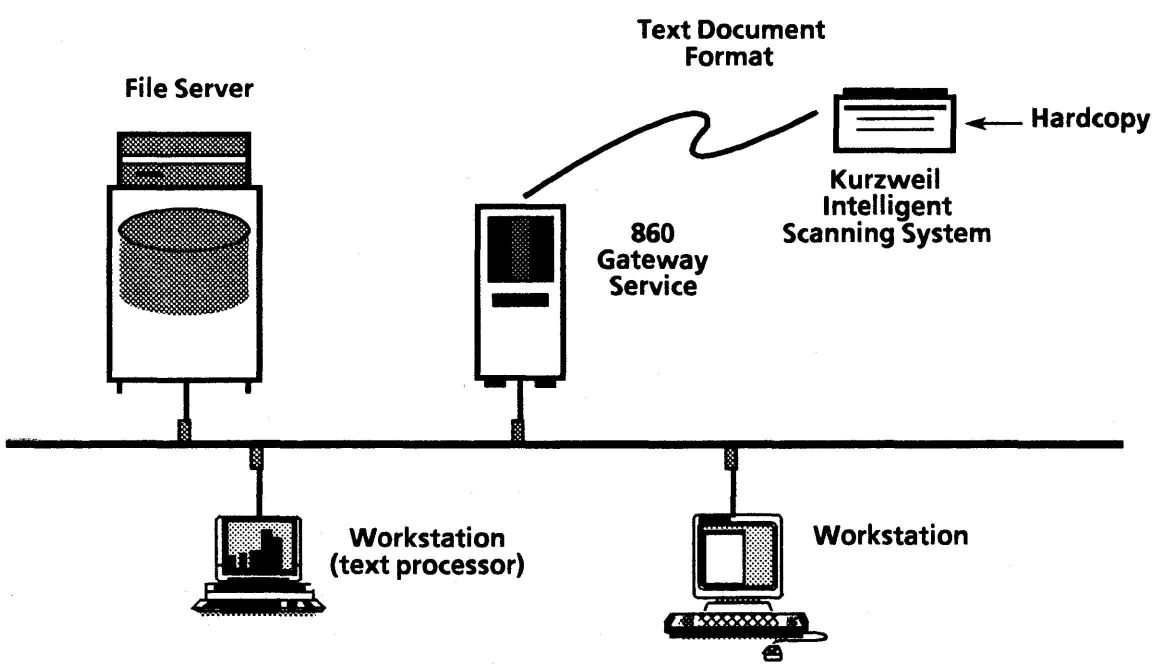


Figure 11-3 **Intelligent scanning system converts hardcopy to text**

Electronic reprographics

Hardcopy is normally reproduced by conventional copiers and duplicators or printing processes which use optical imaging systems. An alternate way to reproduce hardcopy is to electronically scan it and then print the scanned image on an electronic printer. Such a capability could be made available with any scanner or printer that can be connected together such as the Xerox 150 GIS and 9700 EPS. However, the lower speeds of scanners, lower resolutions (300 spi), and higher costs of such devices today make the reprographic application less practical.

With the availability of lower cost, high-resolution, and high-speed electronic scanners and printers, the reproduction of hardcopy by electronic means becomes not only practical but offers a user a number of important capabilities and productivity benefits. The following are only a few of the many benefits of electronic reprographics.

Input/output flexibility

Original documents have to be scanned only once instead of being fed through a recirculating document handler or requiring the use of output sorters, since the original can be scanned and stored, then printed in collated sets. Electronic reprographics provides great flexibility in the handling of both input and output, including variable page sizes, one-or-two sided impressions, and light impressions or dark backgrounds.

Demand printing

Frequently needed documents can be stored on the reprographics system or at an XNS file service when they are scanned and printed on demand. This saves the time spent in finding originals from a conventional file.

Signature printing

Commercial printing industry documents are often printed in 2-up (or n-up) signature format, where two pages are printed on each side of each piece of paper (see Fig. 10-4). With conventional reprographics, this is a long and laborious process which involves reducing the masters, arranging them in order (different from page number order) and then pasting them in place. Frequently the final product is a second or third generation copy resulting in loss of quality. With electronic reprographics using Interpress, this process is straight forward and can be performed almost instantaneously by the computer, without errors or loss in quality.

Custom publishing

With electronic reprographics, it is easy to add customized information such as a person's name and address at the time of duplication (sorted by zip code if necessary). The mailing lists may be electronically stored and maintained at the duplicator or transferred from an XNS workstation via the network when needed. Other types of custom information, such as time/data stamps and page numbers, may be similarly added automatically.

Quality improvements

A document in electronic form is easy to manipulate and correct. It would provide great flexibility in many quality improvement operations such as removing unwanted information or spots, increasing sharpness of lines, making image darker, and improving contrast.

Combining information forms

Availability of high-speed, high quality scanning will make it easy to combine hardcopy and electronic information in imaginative ways. Such a system would bridge the gap between conventional and electronic office information systems.

The XNS architecture makes this capability very versatile indeed. The scanned documents are encoded in RES and compressed, using the best available compression technique. These documents can be manipulated and printed using the power of Interpress and print services, filed using XNS filing, distributed with XNS mail, edited at a workstation, or sent to any device that is directly or indirectly connected to the internet (including remote facsimile machines).

Xerox Network Systems architecture is a general-purpose network architecture with a complete set of basic tools for accomplishing nearly any objective that involves a distributed information system. Xerox has also provided XNS functions particularly suitable to the applications in which Xerox is especially interested and for which Xerox has major product offerings. These applications include office information systems, computer-aided design, sophisticated computer language-based resources, and electronic publishing. All of these applications and products benefit from the underlying XNS services and network capabilities. This discussion of XNS applications focuses on the XNS system elements and how they can be combined to provide solutions to customer problems. Architectural details can be inferred from the previous discussion on standards, protocols, and services.

Office information systems

Ethernet and Xerox Network Systems have traditionally been most closely associated with mainstream office information system applications. XNS systems are installed throughout customer facilities, interconnecting different groups of people with diverse responsibilities. Separate, closely located facilities are interconnected to form "campuses," and individual campuses are linked with remote locations by means of conventional telecommunication circuits or public data networks.

Throughout such facilities individual work progresses with the assistance of a number of high-powered workstations, servers, and terminals. Although many of these are made by Xerox, not all need be; the system enables the direct interconnection of IBM personal computers and the DEC VAX "superminicomputers," for example. Mainframe data processing systems are connected via gateway services which provide the bridge to other network systems such as IBM's SNA.

Documents are created, edited, merged, filed, printed, and reviewed; electronic spreadsheets are used in various management decision processes; programmers write software for still more applications; individual data bases are built, maintained, and linked to mainframe data management systems; and individual users are interconnected by means of a global electronic message system. These and many other applications

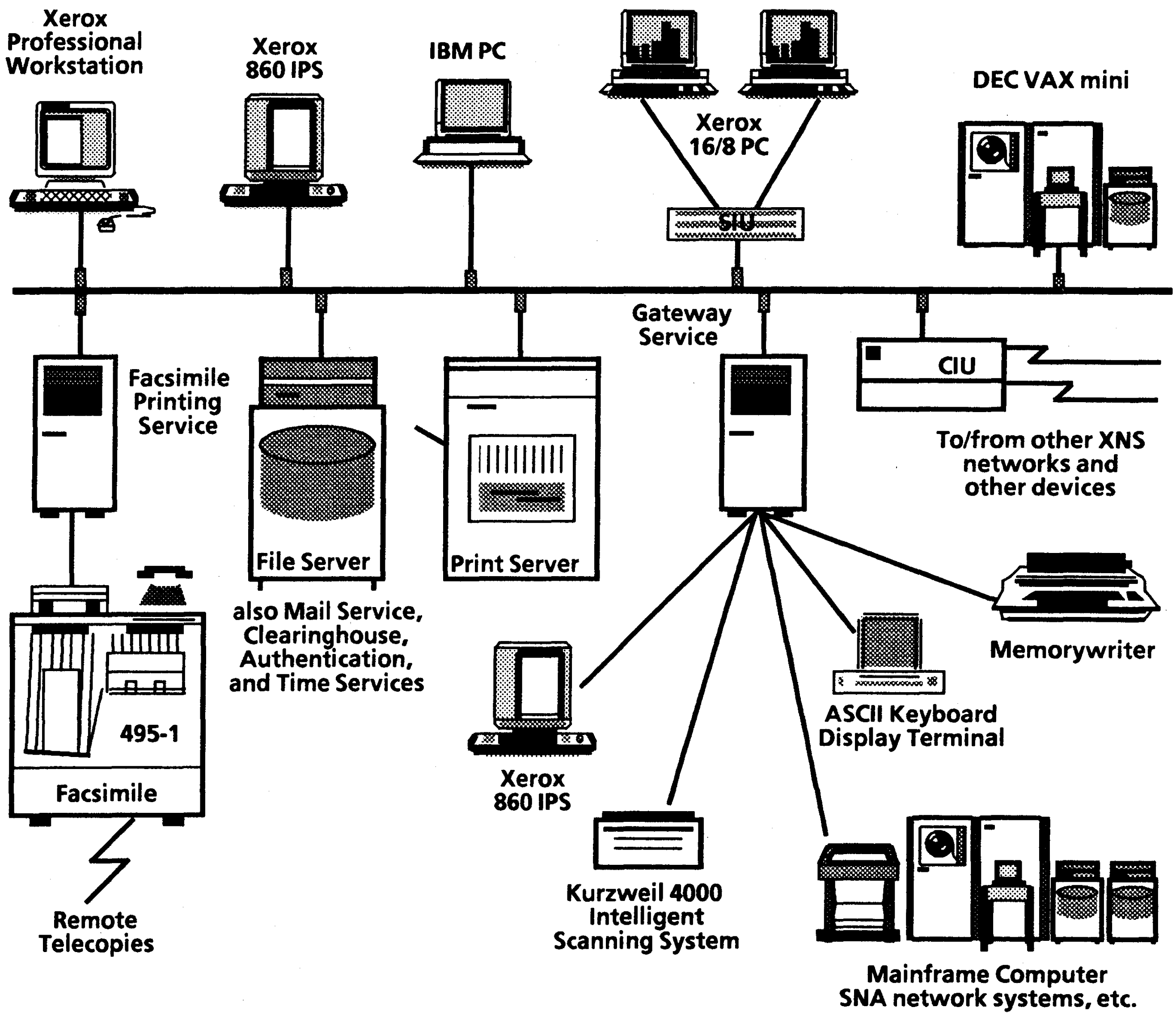


Figure 12-1 **XNS applications in office information systems**

are routine uses of XNS-supported office information system products.

Fig. 12-1 shows the array of equipment that makes this possible. The backbone of such an arrangement is the XNS local area network, Ethernet, which is wired throughout the facility and to which all the system elements directly or indirectly attach.

Other system elements include:

Professional workstations

Xerox professional workstations such as the Xerox Star are among the most sophisticated workstations presently available. They have pioneered a number of concepts now widely accepted throughout the industry including a large, high-resolution display based on "bit-map" (also called "all points addressable") techniques; a "mouse" for a better user interface; and a software-alterable keyboard whose functions can be modified to suit different applications (such as entry of text in other languages, mathematical formulas, etc.).

One of the most important aspects of these workstations is their uniquely powerful user interface which provides up to six simultaneously-available windows for concurrent operations in as many different applications, a property-sheet approach to the entry of control parameters, and the extensive use of "icons" for identifying system resources, data entities, etc. These workstations provide an extensive repertoire of standard software, including text creation, document preparation and layout, illustration and generated graphics, a powerful tabular mechanism for arranging data, and complete access to the various XNS application services.

These resources combine to make the Xerox professional workstations some of the most powerful and versatile workstations presently available, capable of supporting a vast array of user applications: multilingual document creation and composition, spreadsheet and other numerical operations, records processing, forms design, special graphics effects, electronic mail, filing, printing, publishing, and access to mainframe data processors via 3270 emulation.

Text processors

The Xerox 860 Information Processing System is one of the most versatile text processing products available. When connected directly to the Ethernet, that versatility is combined with access to the sophisticated networking resources of XNS. The 860 features a choice of screen sizes including a full page display, powerful word processing software, document layout software, mathematics programs for data manipulation, records processing, spelling correction, sort programs, the ability to handle foreign language fonts and other constructs and a wide variety of utility software related to specific user environments, such as law, medicine, engineering, etc. When interfaced with the XNS network, the 860, like the professional workstations, can use the network's extensive filing capability, its shared electronic printers, its mail system, and its access to external systems such as large computers.

IBM Personal Computer

The IBM PC has been widely adopted for a variety of personal computing tasks. With the IBM PC connected to the XNS network through an Ethernet interface contained within the PC, those same applications are now enhanced by access to the XNS filing, printing, and mail functions. Special software provided by Xerox operates in the PC to give the user compatible access (via Courier remote procedure calls) to these functions. Straightforward methods are used, ensuring that the IBM PC user will be able to use the XNS resources like an XNS workstation.

Xerox personal computers

The Xerox 16/8 combines the "best of both worlds:" eight-bit CP/M computing and 16-bit MS-DOS (IBM PC operating system compatibility) computing. These two operating systems give a user access to both of two important personal computer software repertoires: that written for CP/M-based machines, such as the Xerox 820-II, and the major portion of that written for the IBM Personal Computer and compatibles. The Shared Interface Unit provides the physical connection of one or a pair of computers to the Ethernet; the connection can be local or over a telephone circuit at up to 9600 bps. This means that whether the 16/8 is remote or co-located with the XNS network, it can use the XNS application services, including filing, printing, and mailing, like any XNS workstation.

Intelligent character recognition

The Kurzweil 4000 Intelligent Scanning System may be connected to the network via the 860 Gateway Service. This allows a user to input almost any typewritten or printed hard copy into the system as text characters without having to manually rekey the document. This is a great productivity aid in offices where large amounts of data or documents need to be transcribed.

Electronic typewriters

The Xerox Memorywriter models with the communication options may be integrated with the system via the Interactive Terminal Service. Basic mailing service is available to even the lowest model, the 610 C1. Some of the higher models, which have more internal storage, may also use the filing, printing, and document interchange functions. Electronic typewriters are useful as an inexpensive document creation device.

Non-XNS workstations and terminals

Any personal computer, workstation, or terminal that supports the standard asynchronous ASCII communications can connect to the network via the Interactive Terminal Service and send or receive mail messages and documents. This allows people to access information from outside their office, whether at home, in a hotel, or in another work location.

Facsimile devices

Facsimile devices, such as the Xerox 495, can be used as inexpensive local printers, or as a means to transmit print images to remote facsimile devices. The facsimile technique is ideally suited to printing the complex text and graphics documents which XNS workstations are capable of generating. XNS workstations can send documents directly to any Xerox or non-Xerox facsimile device that is compatible with CCITT Group 1, 2, or 3 via Facsimile Printing Service.

XNS services

Xerox servers are intended to be configured to match a user's requirements; services are combined with hardware as a function of expected performance and capacity requirements. In small systems, separate functions are typically combined. In larger systems, individual file servers, mail servers, print servers, and communication servers might be found. In Fig. 12-1 several server configurations are shown: a generic server, capable of file service and other functions; a print server, with its printing peripheral attached; and a communication server, with a Xerox 873 Communication Interface Unit (which expands the number of available communication ports). In addition, a DEC VAX is shown—not made by Xerox, but integrated within XNS by virtue of DEC-supplied interface hardware and Xerox-supplied XNS software. Together these servers provide the common services of filing, printing, mail, authentication, Clearinghouse, mainframe communication, and internetwork routing necessary for so many interactive tasks in modern offices.

The qualities of Xerox Network Systems serve these office information systems well. The maturity and completeness of XNS provides comprehensive services while ensuring maximum resistance to obsolescence. The global quality of XNS design makes it possible for offices to be established and automated throughout the world with one logical network integrating all of the individual network systems into a unified whole. Xerox' distributed systems approach to architecture makes it possible for resources to be added gracefully to the system. And, most important, these characteristics are provided to the network users on a transparent basis, leaving them free to concentrate on their professional, managerial, and clerical tasks.

Engineering information systems

In today's fast-paced high technology industries the productivity of the professional staff is one of the most important issues to corporate success. Command of a developing market will often go to an organization that is first to complete a new design or a critical design adaptation. This means doing development work quickly and doing it right the first time.

XNS makes possible an engineering support system that combines the best features of highly specialized automated engineering design with the comprehensive office automation functions previously described. An example of such a system features the Versatec Expert Electronic Design Workstation as the primary engineering support tool with the other elements of Xerox' office information systems providing backup functions. Together these elements greatly increase the precision and productivity of engineers engaged in electronic circuit development.

Fig. 12-2 shows such a system. The backbone of this system is an Ethernet local area network interconnecting a variety of workstations, terminals, and servers. All of the functions and services characteristic of a modern automated office are also provided. Additionally there are the specialized functions of the Expert system. These include:

Engineering documentation

Recognizing that a significant part of an engineer's time is spent preparing documentation, the design of the Expert

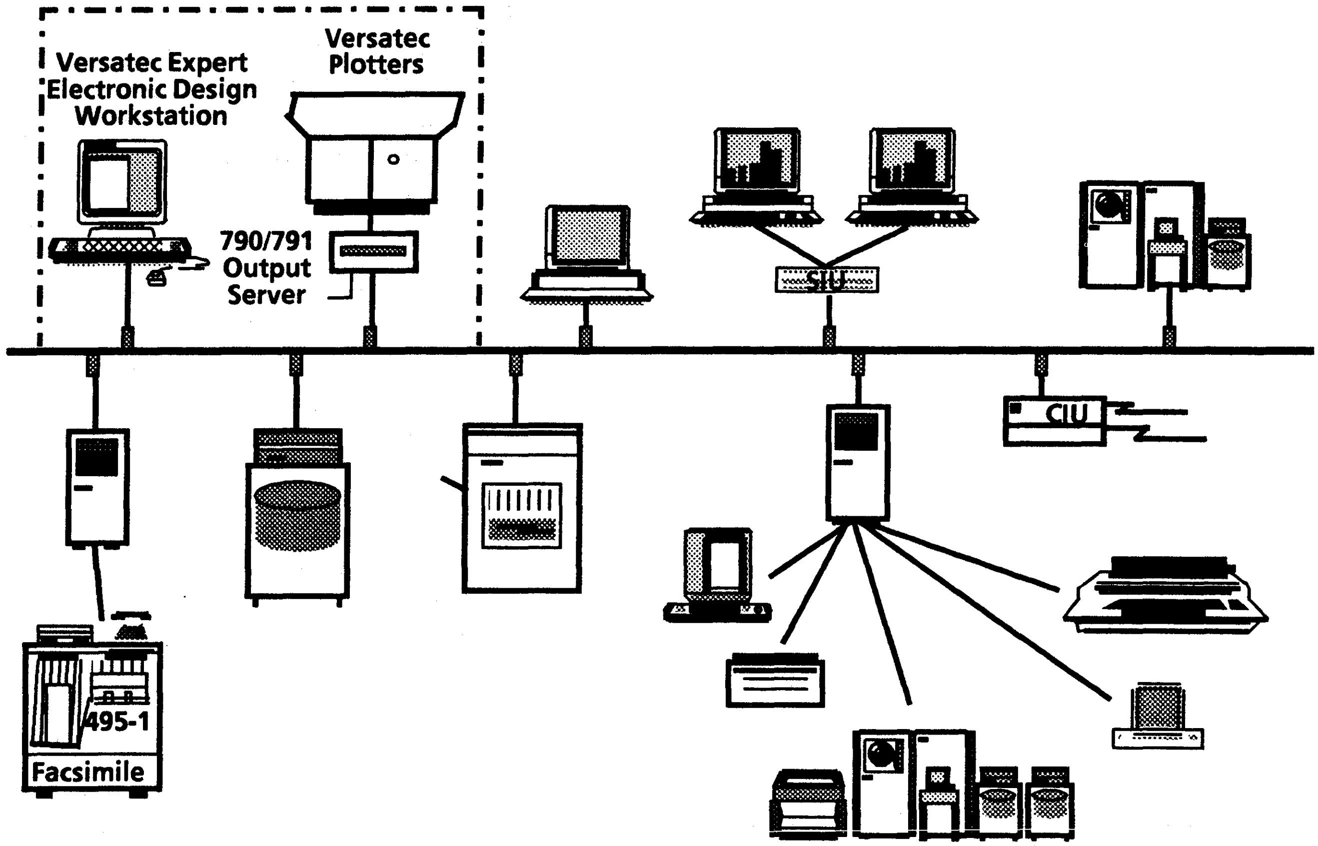


Figure 12-2 **XNS applications in engineering information systems**

workstation permits it to function as a complete Star Information System, with its powerful document creation and editing features.

Schematic design

Using the Expert workstation, an engineer can create and edit logic and similar diagrams unconstrained by conventional office document format considerations.

Logic simulation

The same Expert workstation used to perform logic design also permits the design engineer to simulate the behavior of logic circuits. With both processes on the same workstation and operated by means of a consistent user interface, the productivity of the overall development effort is greatly increased.

Component placement

After the design has been made and simulated, the engineer can use the Expert workstation to create the physical realization by interactively establishing circuit board component placements and conductor routings.

Drafting

Finally the physical aspects of the circuit board can be documented with the Expert workstation acting as an automatic drafting station.

Once the electronic design work is complete, the rest of the XNS system elements support continuing engineering responsibilities. Reports, specifications, test procedures, procurement documentation, standard parts libraries, etc., are filed, retrieved, exchanged, printed, and updated using XNS functions.

Where special engineering documentation is required (e.g., large format schematics and board layouts), Versatec widebed and color plotters are interconnected with XNS via the Versatec 790/791 output servers. Use of Interpress in the XNS architecture makes it possible for even documents of great size and unusual graphic complexity to be handled in a manner entirely consistent with the way other documents are handled. Integrating these special professional applications into the basic set of office services is an example of XNS adaptability.

Programming and knowledge-based systems

One of the most exciting aspects of information systems technology is the growing availability of highly structured software development facilities. These facilities combine hardware and software that permit programmers to work more effectively on projects of increasing scope and complexity. Xerox has pioneered the use of such facilities, supporting large programming staffs and diverse software design projects with one of the most sophisticated sets of development resources in use anywhere. At the same time, Xerox research groups have pioneered the growth of artificial intelligence applications, through specialized adaptation of those same resources and with the help of special computer languages.

Underlying all these efforts have been the facilities of Xerox Network Systems, providing many of the same functions and support for advanced computing in office automation and other applications.

Fig. 12-3 provides an overview of these sophisticated computer language-based resources supported by XNS; each is discussed below.

Xerox Development Environment

There was a time when a programmer's set of tools consisted mostly of a flow chart template, a pad of coding sheets, and a deck of punched cards containing source language statements.

Today's programmer cannot work that way. For one thing, the programs written are more complex than they were before. For another, the machines on which they are intended to run are more complex. Often the programs are written to interface

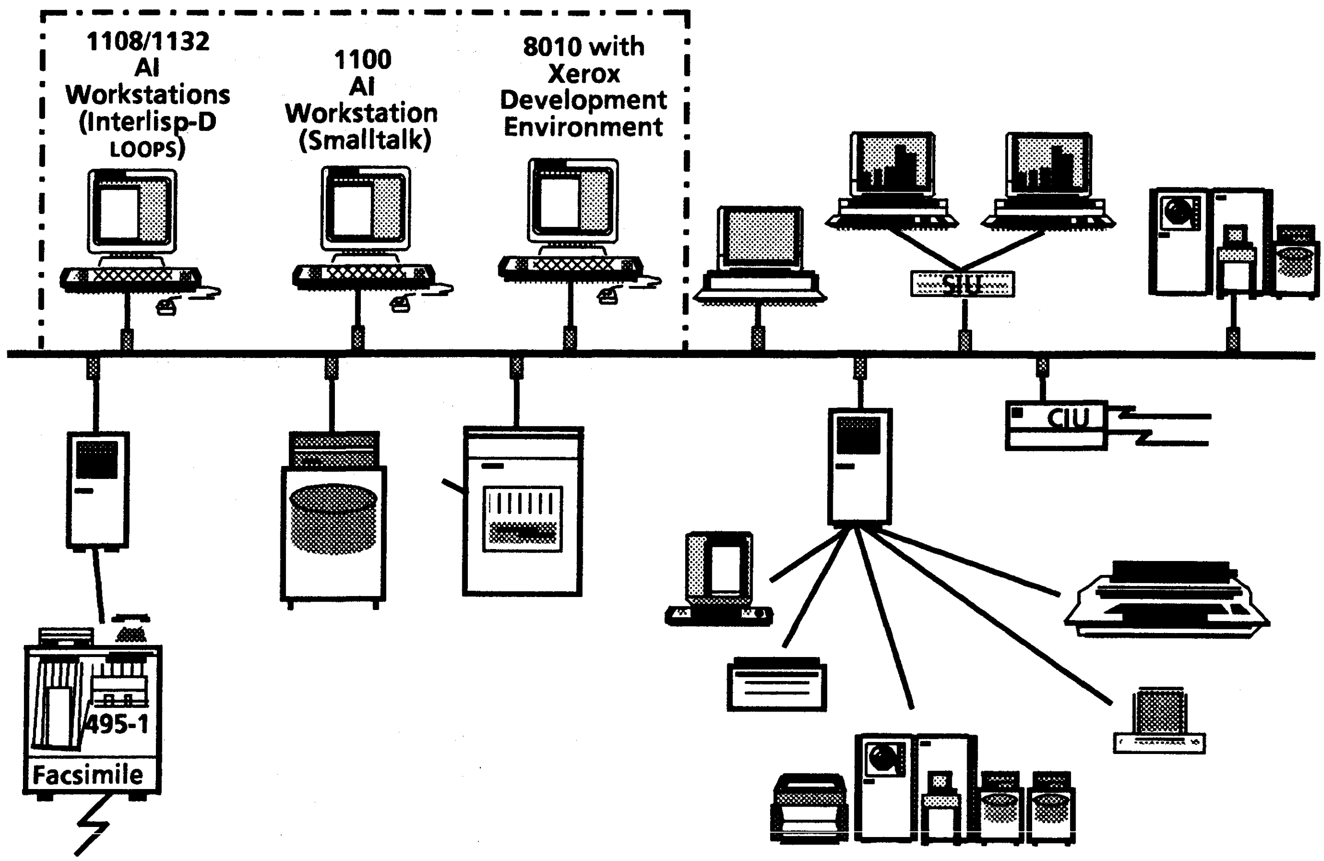


Figure 12-3 **XNS applications in artificial intelligence and software development**

interactively with human beings—in some respects, the most difficult challenge for software design.

Programming productivity and quality

As a result, new tools are being made available to aid programmers, by improving accuracy, productivity, and overall quality of program design. These tools are partly an outgrowth of the new challenges facing programmers, but they are also a result of the availability of very powerful personal workstation technology.

An example of such a system of tools is the Xerox Development Environment (XDE), a combination of hardware and software aimed at improving programmer productivity and programming quality. XDE is an organized set of software resources packaged to run on the same hardware that supports the Star Information System. XDE can be thought of as the Star equipped to support programming activities, much as the Versatec Expert system can be thought of as the Star equipped to support engineering design activities.

Highlights of the Xerox Development Environment include:

- Tools for software development in Mesa, the Xerox-developed programming language: compiler, binder, debugger, performance measurement facilities, etc.
- Tools for software development in commercial languages (e.g., C, Fortran) will also be available.
- A project management system for supporting program development (in any language).
- An operating system (developed by Xerox) intended to support real-time interactive applications. It includes virtual memory management, process management, interfaces to the network services, etc.
- Application tools for general end-user support (the XNS services): mailing, filing, printing, external communications, etc.

Using XDE, programmers are able to use XNS resources to support program development and then *integrate XNS services into the programs under development for the end-user to use.*

The entire focus of the Xerox Development Environment is on the dual objectives of productivity and software quality; important enabling resources for this are the underlying services and protocols of Xerox Network Systems.

Artificial intelligence applications

Xerox' work in artificial intelligence (AI) systems is taking place at such diverse locations as the Xerox Palo Alto Research Center, Xerox Special Information Systems in Pasadena, California, and other locations in Japan, Western Europe, and Canada.

Artificial intelligence systems are designed to employ a knowledge base and a set of inferential rules that relate to a specific field of endeavor, such as medical diagnosis or geological strata analysis. The key to AI systems is their ability to retain the results of early processes, allowing those to influence the outcome of subsequent processes.

One of the most important aspects of artificial intelligence is the development of "natural" computer languages, which are very much more human-like than conventional computer languages. Understanding and processing such natural languages is an application of the principles of artificial intelligence.

AI is regarded as one of the most promising fields of computer science research, seeming to hold many of the keys to bridging important gaps between the world of digital machines and the far more complex real world.

At present, much of the ongoing AI work is in exploratory development, rather than production applications. Many of the requirements of AI development parallel those of programming development everywhere: extensive processor hardware resources, adequate secondary storage and file service, flexible printing capabilities, and a message system capable of interconnecting a programmer with his peers. Such a facility is offered by XNS, as the AI workstation products in Fig. 12-3 suggest.

The Xerox AI products include:

Interlisp-D and LOOPS

The 1108 Artificial Intelligence Workstation, which provides powerful processors, large working storage spaces, a bit-mapped display, mouse pointing device, and similar user advantages. Interlisp-D, a Xerox-developed dialect of Lisp, one of the original artificial intelligence languages, is available on this workstation. LOOPS is another software development tool available on the 1108.

The LOOPS Knowledge Programming System, under active research at Xerox' Palo Alto Research Center, supports the advanced AI programmer with an interactive display editor and inspector, a generalized administrative and error analysis system to relieve a programmer from mundane detail, a wide variety of debugging tools, and the Masterscope facility for program analysis.

Smalltalk-80

The 1100 Artificial Intelligence Workstation, physically similar to the 1108 but designed to support the use of Smalltalk-80, a Xerox-developed language. Unlike Interlisp-D, which is a function evaluation system, Smalltalk-80 is object-oriented. The workstations on which these software systems run are part of XNS, which makes it possible for AI programmers to exploit the full range of XNS support. In AI as in every other form of programming development, the results accurately mirror the quality and scope of the tools used to produce them.

Electronic publishing

One of the most interesting and potentially most productive XNS applications is electronic publishing: the creation and reproduction of documents very quickly and efficiently through electronic means. Electronic publishing makes it possible for documents to be produced with all of the quality features of conventionally published documents—special typefaces, integration of text and illustrations, etc.—in a fraction of the time, and often at a fraction of the cost, of traditional publishing methods.

Electronically published documents are comparatively formal, such as reports, proposals, booklets, and newsletters. Typically, these documents are prepared in moderate quantities (compared to a mass circulation magazine or a hardback book) with content whose timeliness is usually very important.

(Note: the term "electronic publishing" has also been used to refer to the fledgling videotex industry, focused on providing magazine- and catalog-like databases for home consumption. Although there are some intriguing interrelationships between the two uses of the term, here it is used to refer to the publication of physical documents.)

Traditional publishing process

Traditional publication of complex documents is an intricate and time-consuming process, as seen in Fig. 12-4. The process is divided into two main parts: document preparation, in which the final form of the document is created, and document production, in which the appropriate number of copies of the document are printed, bound, and distributed.

Document preparation often involves many serial steps. Text must be written, captured by keystroke in some appropriate word processing or other editing system, edited, and input to the document design process. Graphic elements—line drawings, forms, photographs, etc.—must be created, edited, and also input to document design. With all of the elements of the document finally together, the document can be designed, with decisions being made concerning text arrangement, type faces and sizes, the location of graphic elements, and so on. This leads to final form implementations: typesetting, in the case of text, and screening and sizing of graphic elements. These are brought together for page makeup. When all of the pages have been made up, the "masters" can be assembled to form the "camera ready" document.

Document production is based on the availability of camera ready masters. First, the job has to be made ready, in the sense of collecting all the printing and binding instructions, the raw materials, etc. Proof copies are sometimes drawn from the masters for final checking. Then production quantities can be printed, collated, bound, boxed, and placed in inventory and/or distributed.

Electronic publishing process

Electronic publishing has the ability to make substantial simplifications to this complex process. Fig. 12-5 shows how the functional equivalent of all the processes in traditional publishing (Fig. 12-4) can be accomplished with far fewer steps.

Document preparation can, in principle, be accomplished almost entirely electronically. As the author creates the text, software operating in the workstation or available in a nearby

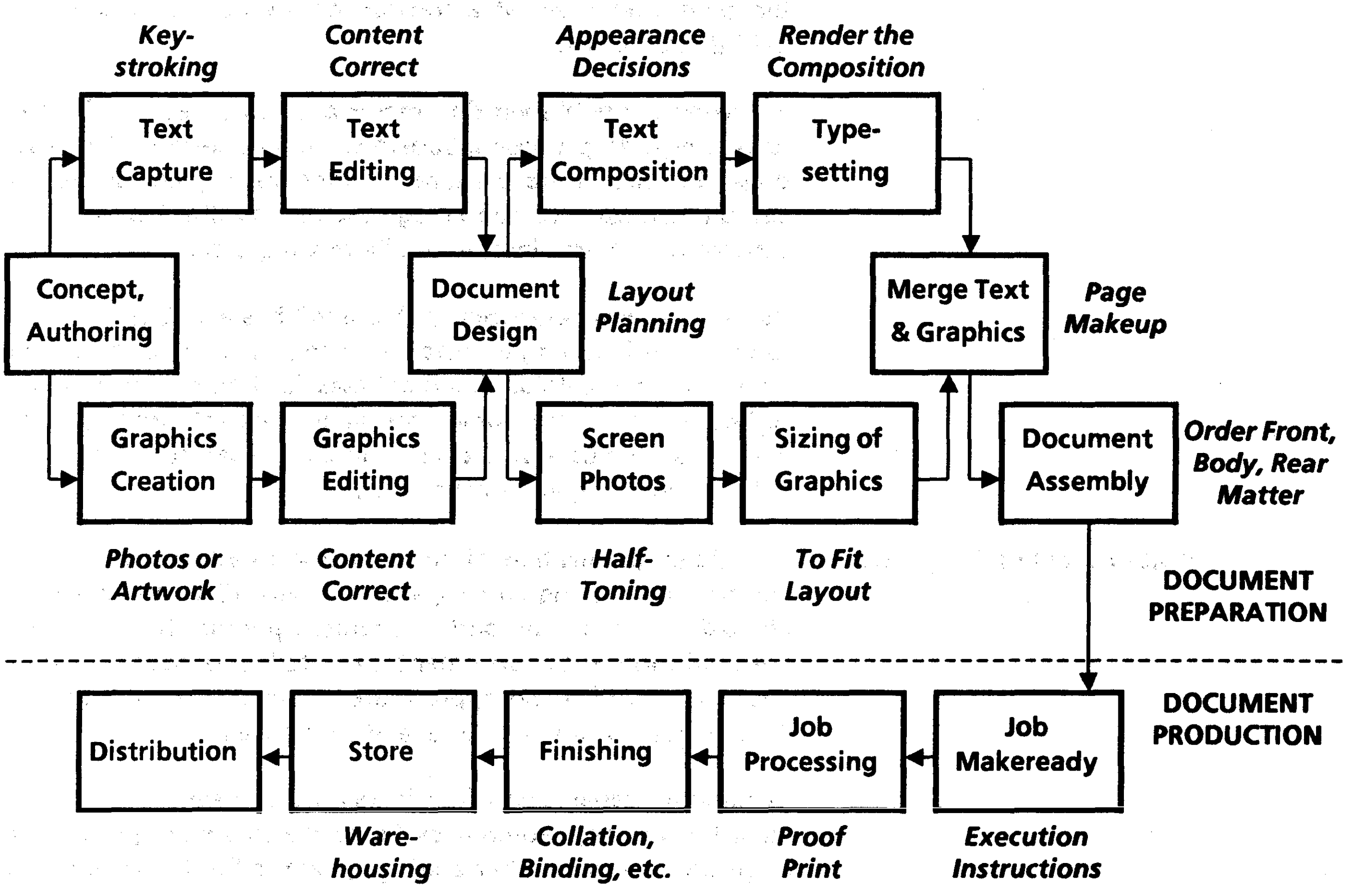


Figure 12-4 **Traditional document publishing process**

specialized subsystem can undertake spelling and grammar checking and a variety of composition and editing functions. This results in completely laid out text—with windows left for the graphic elements. On a properly configured workstation, many of the necessary graphics can be created and edited by the author without requiring intermediate pen-and-ink graphic masters. Where graphic elements are acquired from other sources (e.g., photographs), they can be scanned (converted to electronic form) and subsequently edited. These electronic graphic elements can be automatically integrated with the text to form electronic final-form page masters, ready for production.

In an electronic publishing system, document production can take two forms. One involves the creation of a single original of the document, via electronically-driven typesetting or a high-resolution electronic printer. This original can then be input to a high-volume reproduction process, such as a light-lens duplicator or offset printing press. Alternatively, a set of documents can be printed by electronic printer directly from the electronic master without the production of an intermediate original. This results in a very simple two-step process.

Electronic publishing, as opposed to publishing by more traditional methods, has the obvious advantage of fewer steps,

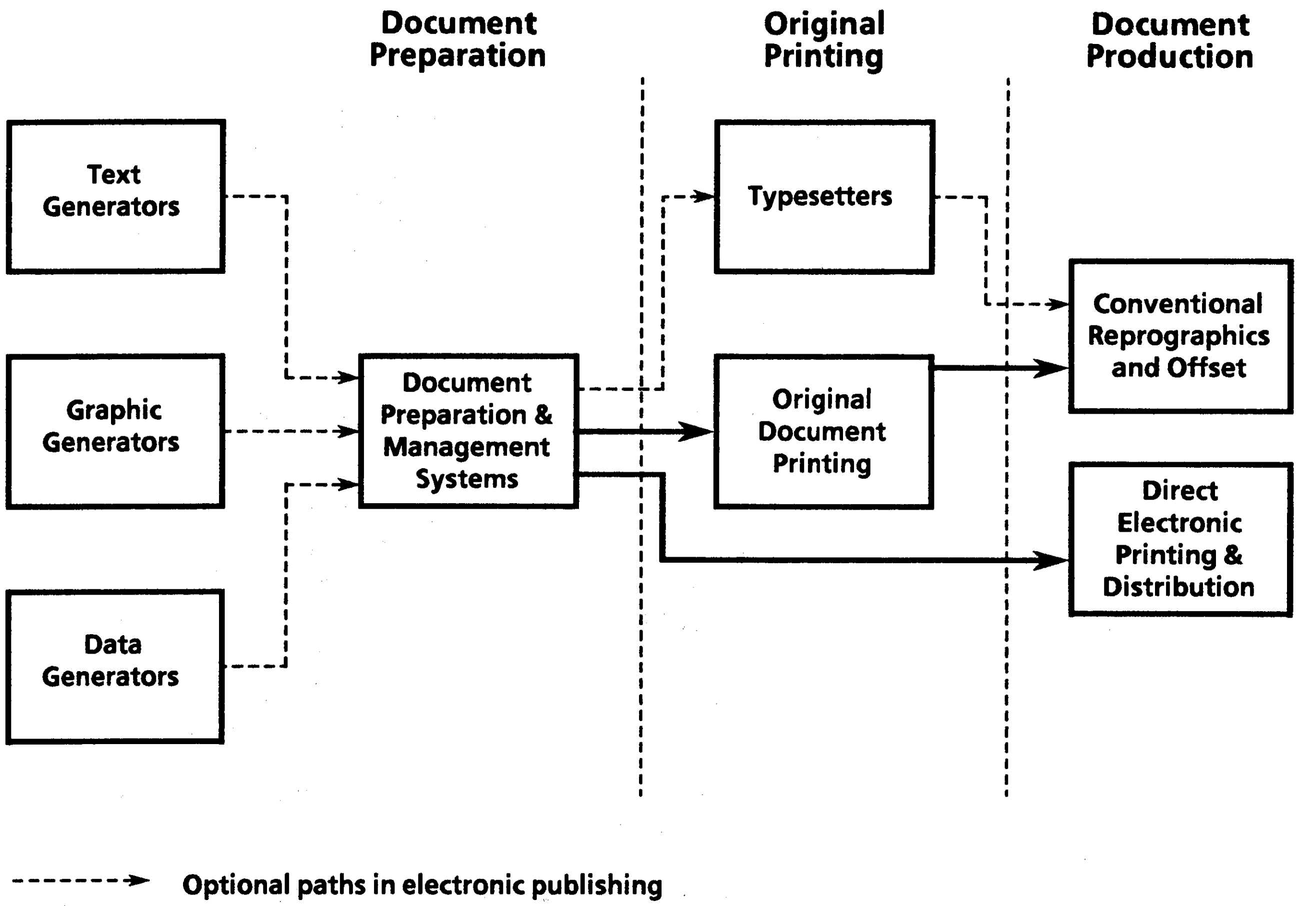


Figure 12-5 **Electronic publishing process**

faster throughput, and higher responsiveness. Inventories can be kept at low levels and documents of all kinds can be published with more up-to-date content. This new technique for publishing also gives the creator and editor of the document a more "hands on" degree of control, which makes it possible to be more creative in the design of the document and greatly speeds the publishing process.

Xerox and electronic publishing: baseline products

Xerox has been active in electronic publishing applications for quite some time, having pioneered with the mainframe-based Xerox Integrated Composition System (XICS) software package. XICS is capable of accepting terminal text input and producing fully composed, made-up page masters for use by host-connected electronic printers such as the Xerox 8700 or 9700.

With XNS, it becomes possible to create the same electronic publishing functionality without the use of a large, expensive mainframe computer—in fact, greatly to transcend approaches using host computers in terms of flexibility, document preparation power, and productivity.

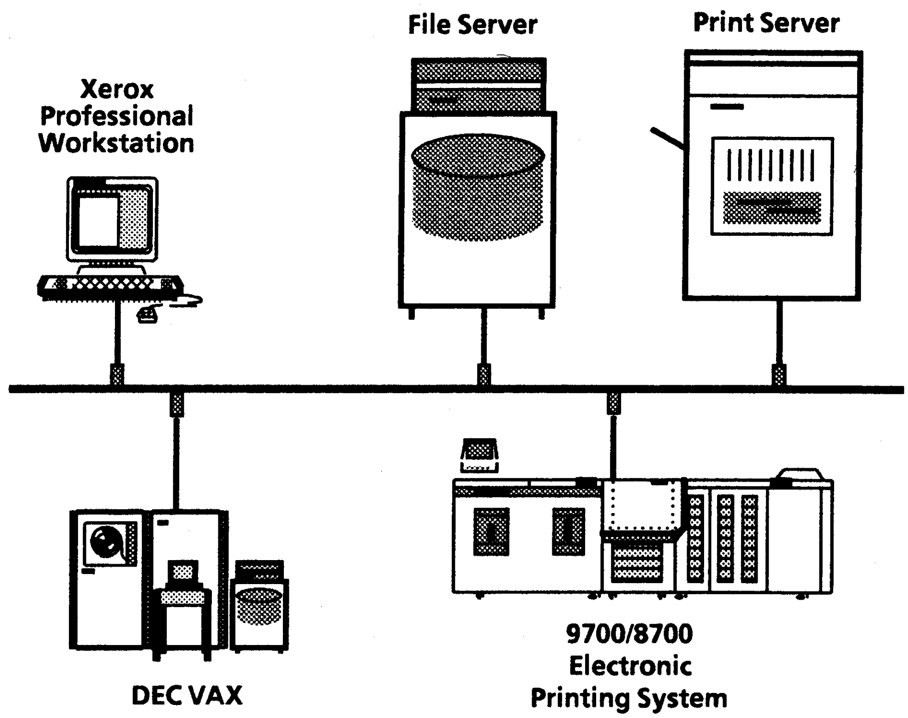


Figure 12-6 **Electronic publishing using Xerox professional workstations**

Interactive publishing systems

Fig. 12-6 shows one approach to electronic publishing, using existing Xerox products. In this example, text and line graphic elements are created on a Xerox professional workstation (such as the Star Information System) using that system's extraordinary document creation and handling capabilities. In actual practice there would be a number of such workstations networked together. In some cases a single user would be completely responsible for the creation and publication of a given document; in other cases more than one author would contribute to the overall effort; and in still other cases the document would be assembled from archival (filed) material, plus new material from one or more workstation users. In any case the interplay between the workstations and the file servers would be substantial, as the document text and graphic elements were input, exchanged between workstations for review, edited, and finally assembled into a single document. That document would ultimately reside in a file, ready for document production.

As indicated in Fig. 12-6, document production could take place using the relatively low-cost 8000 NS print server which operates at speeds up to 12 pages per minute, or using a much larger 9700 Electronic Printing System which operates at speeds up to 120 pages per minute. Other Xerox printing systems are available to function as print servers at intermediate speeds and costs. Whichever printing system is used, the sophistication of Interpress as a document representation technique makes it possible for a wide variety of complex print artifacts to be

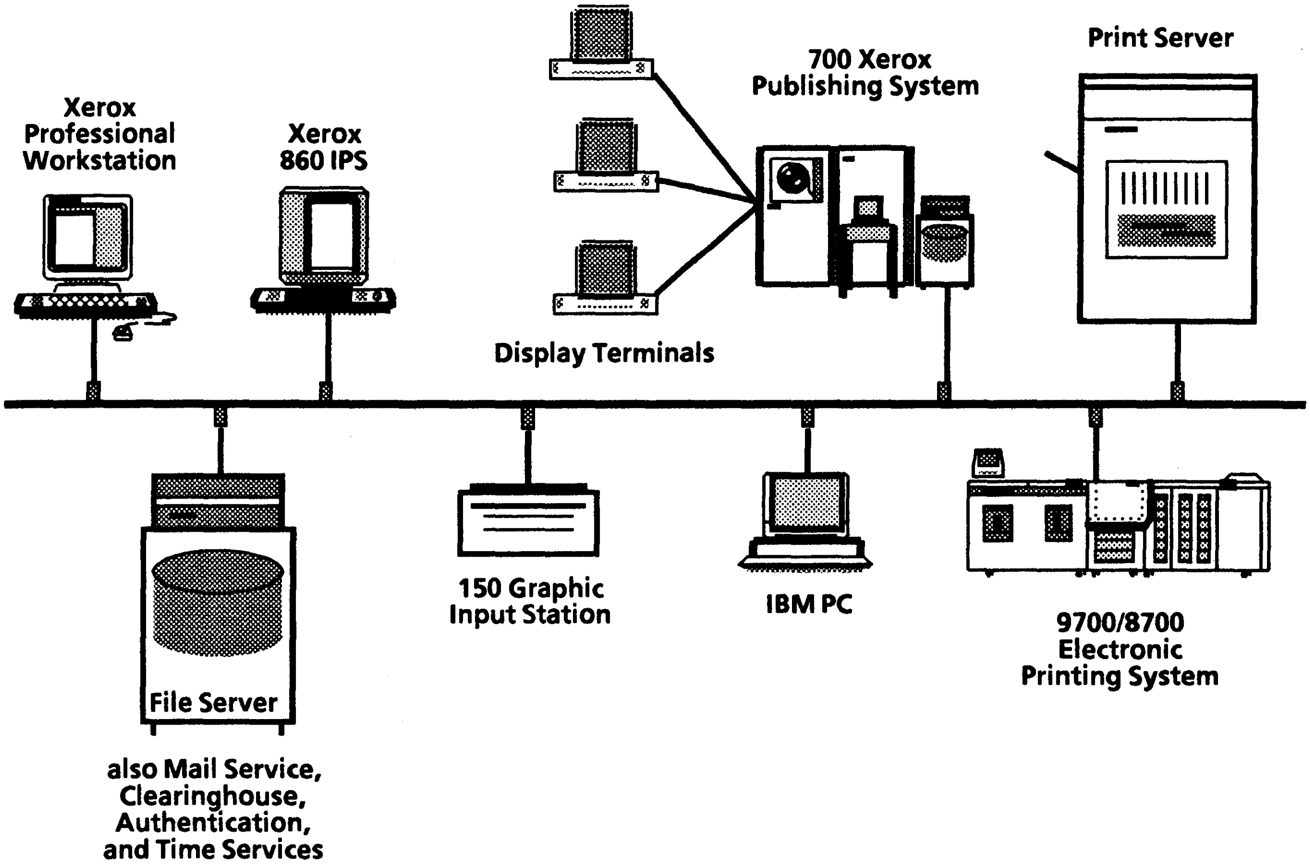


Figure 12-7 **Electronic publishing using the 700 Xerox Publishing System**

produced: multiple fonts, rotations, merged text and graphics, "two-up" printing of multiple pages, etc.

This approach to electronic publishing is inherently user-driven: an individual, using a workstation keyboard and display, initiates and directs the process, interacting with the system at each stage of the process to review results and make decisions.

Batch-oriented publishing systems

An alternative approach, suitable in a number of situations where the interactive approach is inappropriate, is shown in Fig. 12-7. Here the central element is the 700 Xerox Publishing System (700 XPS). The 700 XPS is a fully-featured large scale minicomputer to which a number of terminals may be connected, and that interconnects with the network as shown. Text is input from the terminals or over the network from such workstations as the 8010, the 860, the 16/8, andthe IBM PC.

Graphic elements such as photographs and line drawings can be introduced to the system by means of the 150 Graphic Input Station, a scanning peripheral oriented to publishing systems. The 150 scans each document, converting it to a digital representation that is properly scaled to suit the intended location in the final document. The 150 also includes user controls to produce a halftone effect with photographs, rendering them suitable for printing by xerographic techniques.

The 700 XPS is a batch-oriented composition system, permitting highly efficient handling of standardized document types.Through the use of style sheets, which define the desired output characteristics of a document (relative to such matters as text formatting, the inclusion of graphics, pagination, etc.), the 700 XPS accepts the text input and automatically generates final-form masters for use in electronic printing, typesetting, and other production processes. Unlike the preceding example, the user—particularly the one responsible for inputting the text—does not have interactive decision making over the design of the document. Instead, this outcome is predetermined and enforced by the 700 XPS.

The duality shown here emphasizes the flexibility of Xerox' approach to this developing market and application, and underscores the power of the XNS architecture. Whether batch- or interactive-oriented, and despite the myriad sources of input information and destinations for final-form output masters, XNS makes it possible to create systems which meet a user's requirements.

Xerox and electronic publishing: potential extensions

NOTE: The material in this section is provided for the sake of illustrating characteristics of XNS; not all of the attributes described are presently available. This discussion is not intended to represent future product commitments by Xerox.

Although the systems described above provide fully capable electronic publishing services, the state of the electronic publishing art is evolving quickly. The future will bring a number of important extensions and enhancements, each facilitated by the flexibility and open-endedness of Xerox Network Systems.

Increased integration

One of the most obvious extensions to the Xerox electronic publishing system is increased integration: bringing more sources of input text and graphics into the system.

Fig. 12-8 shows how the features of 8010- and 700-based electronic publishing can be combined, along with additional elements from Xerox' office automation product line. In this scenario, text can be created locally by means of the 8010, the 860, the IBM Personal Computer, the Xerox 16/8 Personal Computer, and the terminals attached to the 700 XPS. Text created at a distance from the network, such as the 860, ASCII keyboard/display terminals, the Xerox Memorywriter, or even large-scale mainframes, can be input to the system by means of the Gateway Service.

As before, graphic elements can be introduced to the system by means of the 150 Graphic Input Station.

Once in the system, text and graphic elements of the anticipated publication can be stored and manipulated in a variety of ways. The 8000 NS File Server can store these elements as they are being accumulated, and in intermediate stages of their processing. Potentially, the 700 XPS can also act as a file server for this same purpose.

Text elements can be created on one device (e.g., an IBM PC or the Xerox Memorywriter) and edited on another device (e.g., the 8010 Star Information System) through the facilities of XNS' Document Interchange Service, which uses the Xerox Interscript document description language.

Similarly, graphic elements input at the 150 or delivered to the system through the communications server will be able to be viewed on a 8010 screen, or an editing screen associated with the 700 XPS. Still further extensions to this technology will make it possible for those graphic elements to be edited: content can be changed, the shape of the elements can be changed, extraneous material can be removed, text annotations can be added, etc.

The full composition services of the 700 Xerox Publishing System can be brought to bear on these diverse forms of information, combining them to produce the final document, then managing the assembly and production of the document by means of appropriate print servers.

Future print servers will lend themselves to this process by providing enhanced image quality, special document handling features, and continually decreasing per-page costs.

Throughout this scenario of wider integration, the characteristics of Xerox Network Systems makes possible the graceful expansion of the network to include new elements and provide conversion and communication paths between them.

Enhanced interactive publishing

As indicated above, the 700 Xerox Publishing System is intended for batch operations, where the details of the resulting document format and structure are predetermined. The 700 responds to preprogrammed directions for the production of published documents.

By contrast, the 8010-based approach is interactive. A user may interact directly with the page composition process. An author has direct control over the details of document form during and after creation of the text and graphic elements that comprise the document content. A number of enhancements are possible that collectively can produce a truly outstanding interactive electronic publishing system, based on 8010-type individual workstations. For example:

- Enhanced editing, permitting more complex page layouts, the use of preprogrammed style sheets for repetitive layouts, etc.

- Handling of scanned graphics, with reformatting and content editing and the integration of such images into documents already containing text and generated graphic elements.

- Extended ability to create graphics, using stored graphic elements and input from computer-assisted design systems (such as the Versatec Expert system, described above).

- Support for larger documents, involving many different authors distributed throughout an XNS network community.

Integrated reprographics

At Xerox, "reprographics" is the general term used to refer to the copying or duplication of already existing material (Xerox' original business, and still a mainstay of its operations). Historically, reprographic products have used "light-lens" techniques: the original document is copied by means of a bright light impinging on it, transferring an image of it through a lens system to a photoconductive surface, and then making the copy.

The traditional light lens technique is not the only way in which this can be done; it is possible to combine an electronic reprographic function with the other elements of a Xerox electronic publishing system to produce an even more versatile facility, one capable of handling not only material created electronically, but material that already exists in printed form.

Fig. 12-8 also shows a reprographic extension to the electronic publishing system discussed earlier. The difference lies in the addition of a special reprographic scanning server, capable of handling volumes of hard copy input material in an efficient fashion. From a network standpoint, this scanner is similar to the 150 Graphic Input Station; both scanning servers produce digital images of the hard copy pages they scan. In actual practice, however, the reprographic scan server will enable new applications because of its high-volume capacity. For example, the scanning server will make it possible to use the exceptional image quality and paper handling characteristics of an advanced print server as a copier or duplicator. Thus, through the interconnections made possible by XNS, a user can save considerable expense and gain greater versatility of available services.

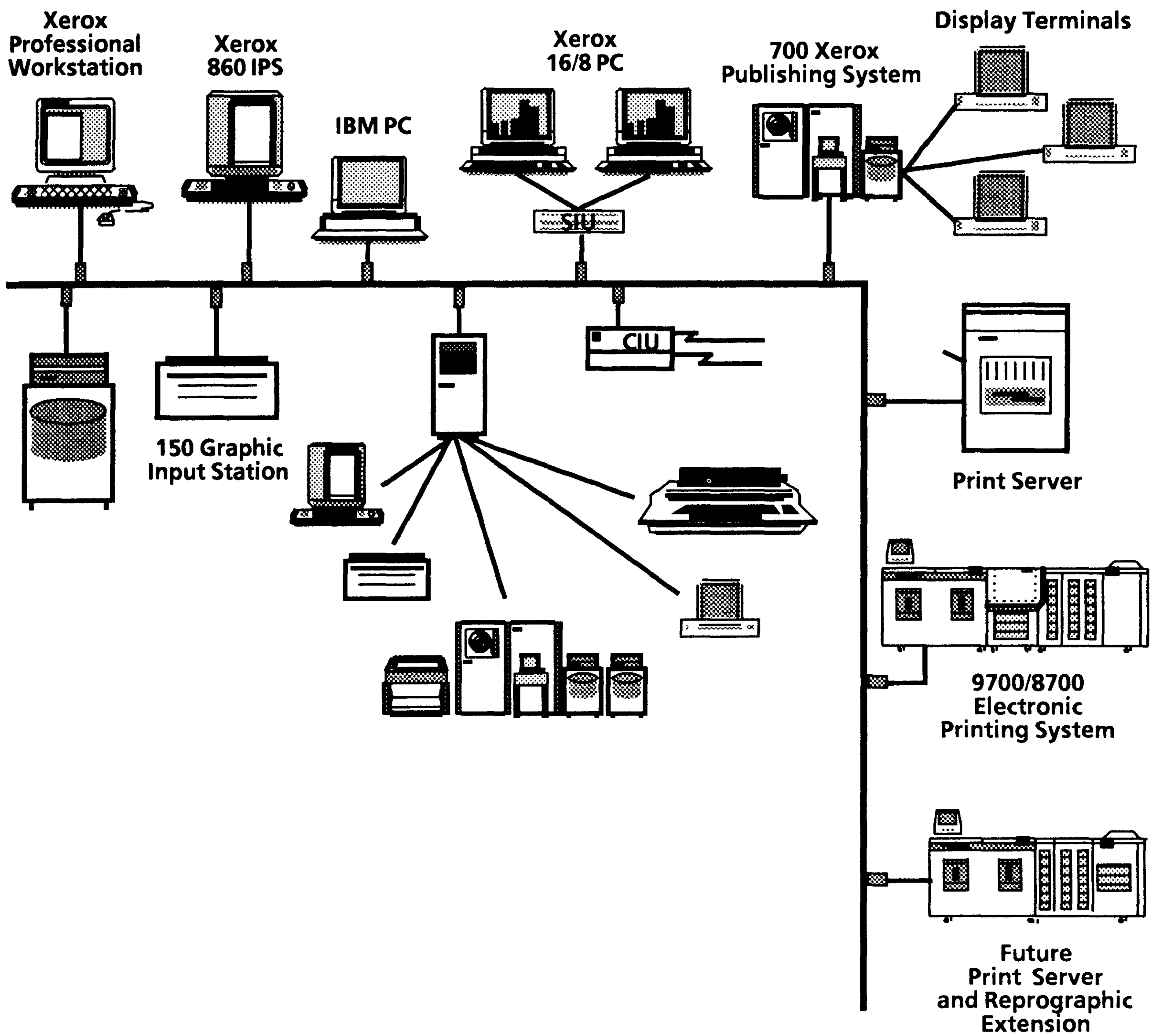


Figure 12-8 **Extended integration in the Xerox electronic publishing applications**

A. OBJECTIVES OF NETWORK ARCHITECTURES

Use of a network architecture in the planning and design of a major information system produces a number of important benefits. Properly designed and rigorously followed, these architectures solve a number of problems that have plagued computer and communication system designers since the beginning of the industry. In the same way, they help overcome a number of limitations which, without a clear architectural foundation, often restrict the value of the system. Examples of the problems addressed by network architectures such as the Xerox Network Systems architecture include:

New communication services

In the early days of data communications the transmission of digital information was usually conducted over ordinary telephone lines, at low speeds, and in simple formats that made equipment easier to design and build. But over the last two decades great strides have been made in data communications technology. Data rates have increased dramatically, classes of services have multiplied, satellite channels have come into common use, entire new forms of telecommunication service have arisen (such as videotex), and more efficient transmission protocols have been placed in use.

One of the objectives of a network architecture is to provide a framework in which new communication services can be introduced *without requiring the redesign of hardware and software not directly involved in data transmission*. Such a provision makes it possible for new classes of communication service to be exploited as they become available, without incurring wholesale obsolescence of the information system design. The network architecture meets this objective by forcing a layered structure that isolates the data transmission functions from the rest of the functions, and by providing rigorously defined interfaces between the layers.

Terminal compatibility

The original communication terminals were patterned after the teleprinter terminals made by Teletype Corporation and others. Slow, rugged, and limited to capital letters, these terminals have been replaced over the decades by an astonishing variety of terminal devices, including:

— Interactive terminals
— Batch terminals
— Personal and business computers
— Distributed processors
— Word processors and other office workstations

—Special-purpose terminals for retail, financial, educational, medical, and other applications.

For each terminal type there are dozens of versions. There are hundreds of terminal suppliers throughout the world, with the number of installed terminals well into the millions. With each new major terminal type introduced, a new set of requirements for compatibility has been created.

A successful network architecture, while not being able completely to solve the problem of incompatible terminals, is at least able to contain the problem to a relatively small number of functional elements within the architecture, isolating the terminal differences to those elements directly concerned with terminal handling. The network architecture provides a structure that makes it possible to perform protocol conversions and related processes that provide compatibility among terminals and other system elements.

Multiple applications

Early data communication networks were usually designed to support a specific application, because terminals and workstations were capable of handling only a limited number of different tasks. Network architectures solve this problem by providing a unified approach to data communications. A single network is capable of handling all the necessary forms of traffic and all the necessary classes of communication service and types of terminals. Thus, the entire cost of a dedicated network is no longer borne by a single application. A network architecture also makes it possible to build adequate redundancy into the information system, enhancing reliability.

Growth and change

Another disadvantage of the single-application approach to network design is that such designs are usually incapable of graceful expansion. Network growth and change is a natural phenomenon, a consequence of corporate growth, new technology, the desire for new applications, and the general trend toward smaller, more personalized information system products (of which the personal computer is a recent spectacular example).

Network architectures solve this problem by providing a clearly stated growth path for the addition of new workstations, new lines, new applications, new classes of communication service, new software modules, and additional features. The modularity inherent in a properly designed network architecture makes it possible to add these elements without requiring major changes to the rest of the information system. Such systems are often referred to as open-ended. Systems designed without the support of such a network architecture are often severely limited when it becomes time to expand or to modify the network.

Distributed intelligence

In addition to increasing the use of data communications facilities, the growth of distributed intelligence—placing the computing resources where a user needs them or where data first becomes available—has changed the character of communications. Where previously most communication systems were designed to relay simple transactions, time-sharing messages, and batch reports, the new requirements include the distribution of software, the synchronization of distributed data base content, and the sharing of distributed resources for program execution. These new purposes change the requirements for such matters as network capacity, response time, and the varieties of data types the network is capable of handling.

Network architectures respond to this new view of data communications by anticipating the variety of data types to be used. A truly general-purpose network architecture views many of the tasks required to support distributed intelligence as the responsibility of the communication system itself. Accordingly, it provides for implementing those tasks within the framework of the architecture rather than leaving them for haphazard implementation elsewhere. Thus program exchanges, remote procedure calls, and global services such as filing and printing are considered part of the overall communication process and are designed into the architecture. By contrast, systems without such comprehensive architectural support relegate some or all of these services to external control, at the risk of uneven or inadequate results, and thus provide far less utility for the user.

System integration

Modern information systems require the various system elements to work together. Integration among system elements (hardware, software, networks) is of particular concern in a distributed/replicative system as opposed to a hierarchical system, such as commonly used by mainframe manufacturers. In the latter, integration is largely accomplished by software exchanges within the main computer at the top of the hierarchy. But in distributed systems, integration is a functional responsibility imposed on each autonomous system element. With no central controller, each element must have enough intelligence to take care of itself on behalf of the user.

Distributed network architectures are designed to support this autonomy. All of the functions cited above, and many others related to the integration objective, are designed into the architecture and its supporting components, permitting design of individual applications to proceed with a minimum amount of concern for the overall behavior of the system.

Local and wide area networks

In any properly designed network architecture, both local and wide area forms of communication can be integrated into the system. The architecture should provide for the requirements of both; one measure of its adequacy and effectiveness is its ability to accommodate the two forms.

Open-ended systems

A network designer cannot successfully anticipate all the purposes to which a user might wish to put a complex multi-purpose information system. This requires that the network must be designed to be open-ended, which means that the architecture itself be capable of being modified as required. This is particularly true in environments such as office automation, where the rules are being written for the first time and are subject to being rewritten. In such systems the eventual destination, from a design standpoint, is less important than the qualities of the system along the way.

An open-ended architecture makes it possible for architectural changes to be introduced with a minimum of disruption to implementations already in place. Strong emphasis is placed on appropriate backward compatibilities so that a user's system is not rendered obsolete, but can migrate to the new state at a time of the user's choosing.

Open systems interconnection

The perspectives and experiences of others are invaluable ingredients in the design of any information system. Accordingly, a successful network architecture must be an open system, capable of incorporating key industry standards (official and *de facto*) as they emerge, and structured and documented in such a way that its own special values can be shared with the rest of the industry. This latter quality is especially important since it helps ensure that others can develop products for and interfaces with what otherwise might be a proprietary network design.

B. XNS AND OTHER NETWORK ARCHITECTURES

A number of network architectures have appeared in the last few years. Some are of a generic nature, typically sponsored by one or more standards-setting organizations; others have been developed by computer system manufacturers and reflect quite specific design objectives. Some of the more prominent of these architectures are shown in Fig. B-1.

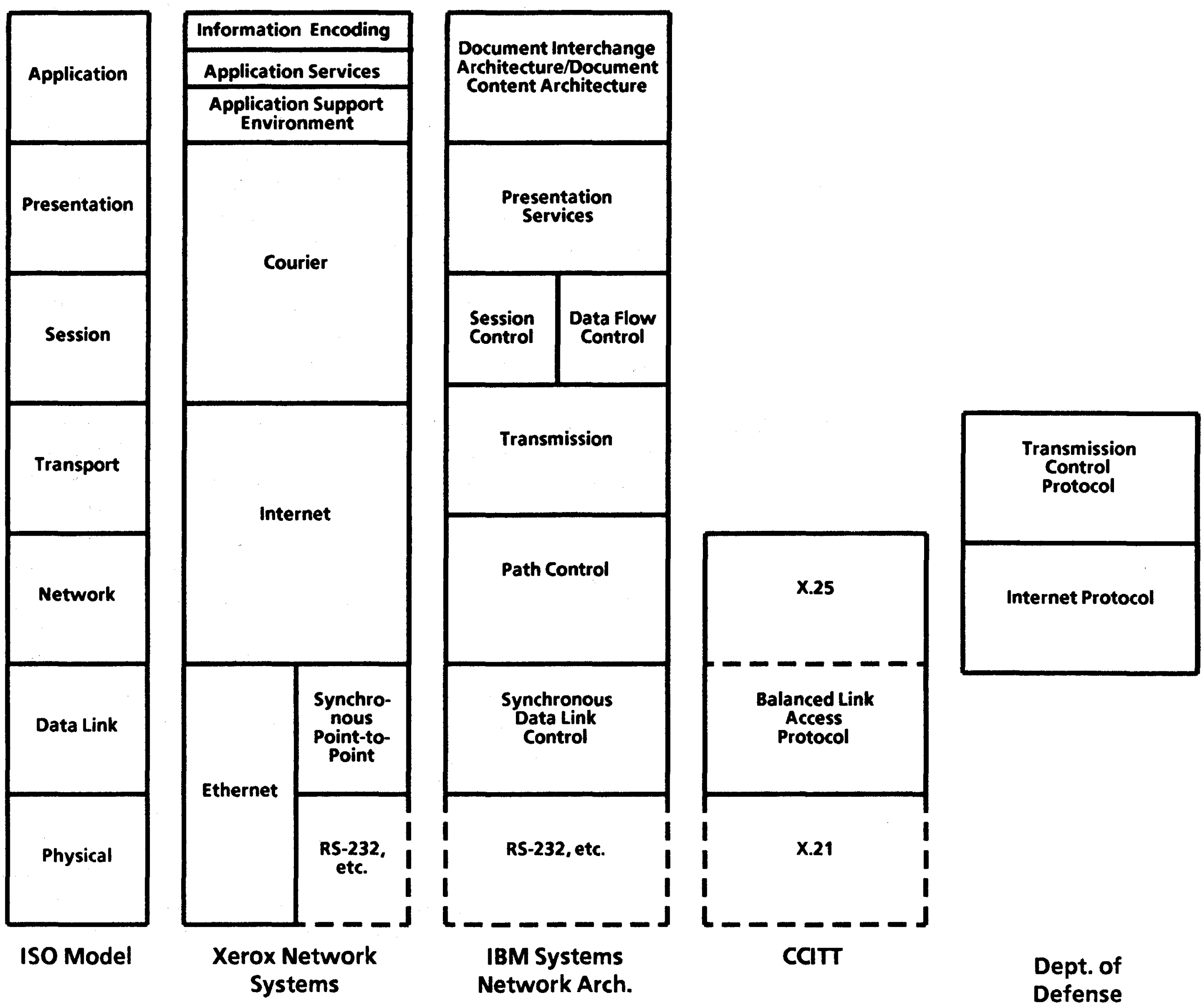


Figure B-1 **XNS and other architectures**

ISO-OSI Model

Fig. B-1 shows the ISO Open System Interconnection Reference Model on the left, with its seven functional layers ascending from the physical to the application layer. Next, the Xerox Network Systems architecture is shown; it extends from the lowest to the highest layer, indicating a comprehensive, multi-purpose architecture. Note the variety of application-layer protocols which highlight the functionality of XNS. The ISO standards so far have only been defined for the lower communications and transport layers, and exclude applications.

IBM's SNA

IBM's Systems Network Architecture (SNA) is shown to the right of XNS. SNA, although oriented to centralized rather than distributed information systems, is second only to XNS as a comprehensive architecture. It provides a set of functions from the data link layer (represented by IBM's link control procedure, SDLC) to the presentation layer. IBM designers do not include application functions in the SNA architecture, but these are represented in the Document Interchange Architecture (DIA) and Document Content Architecture (DCA).

CCITT's X.25/X.21

The CCITT international standards group has announced a set of low-level standards, of which the interface to packet-switched networks, X.25, is probably the most well known. X.25 itself includes both the data link and network layers, but an important subset, the balanced link access protocol (LAP-B), is defined to cover the ISO Model data link layer. Special physical layer interfaces are defined in the international arena; in U. S. applications, the X.21 interfaces are often replaced by the more familiar RS-232 or RS-449 interfaces. Other layers of the CCITT architecture are also being defined.

DOD's TCP/IP

In addition, the Department of Defense, with some assistance from the National Bureau of Standards, has defined an internet protocol (IP) and a transmission control protocol (TCP) that collectively ("TCP/IP") covers the ISO Model layers from network to some of the functions required in the session layer. As in the case of CCITT, TCP/IP is not a complete network architecture, but rather is a set of protocols capable of functional extensions beyond conventional data link operations. Note that no presentation layer functions are available, which means that TCP/IP users are responsible for their own data transformation conventions, a major part of the job of implementing a general-purpose network.

In addition to those architectures whose structures are depicted in Fig. B-1, a number of others are noteworthy. (The references in parentheses are to the ISO layers to which each architecture's uniquely-named functional layers roughly correspond; the ISO layers are shown in Fig. B-1 and explained in section 2.)

Burroughs Corporation

The Burroughs Network Architecture (BNA) , first announced in 1976, is a peer-to-peer, computer-to-computer, and terminal-to-computer network with the following layers: applications, port router (presentation), station (transport), data link, and physical.

Data General Corporation

Xodiac, first announced in 1979, is a network architecture oriented to Data General's Eclipse line of large mainframes, with the following layers: applications, function (presentation, session, and transport), connection (network), link control (data link), and physical.

Digital Equipment Corporation

The Digital Network Architecture (DNA), first announced in 1975, is a computer-to-computer network architecture serving DEC's product line, from personal computers to very large minicomputers, on a peer-to-peer basis, with the following layers: user and network management (application), network application (presentation), session control, end-to-end communications (transport), routing layer (network), data link, and physical.

Hewlett-Packard

The Distributed Systems Network (DSN), first announced by Hewlett-Packard in 1977, implements a distributed computer-to-computer network with the following layers: user language programs (application), network access methods (presentation), network manager (session), message control (transport and network), communication line protocol (data link), and communication electrical interface (physical).

Honeywell Information Systems

The Distributed Systems Architecture (DSA), first announced in 1977, is a host and front-end processor network architecture, with functional layers that correspond directly to the ISO model.

Sperry Corporation

The Distributed Communications Architecture (DCA), first announced by Sperry in 1976, is a host-based hierarchical network architecture for computer-to-computer communications, with the following layers: applications, termination system (presentation), transport network (session, transport, network and data link), and transmission facilities (physical).

Architecture Comparisons

With so many different network architectures available in the market, comparisons are inevitable. Why should a user or supplier adopt the XNS architecture? How does it compare with other leading architectures? The answer simply is: *XNS meets all the objectives of network architectures. No other architecture does.*

The Xerox Network Systems architecture with its long-range view and its global perspective is the most complete, the most comprehensive, and the most general architecture available for integrated information system applications. Further, the architecture is distributed, which means that it is adaptable to the needs of any work situation, and users can grow with it, one system element at a time. At every layer of the architecture, XNS comes out ahead in a direct comparison.

Local area networks

The high-speed Xerox Ethernet solution is very general purpose and adaptable to user needs. The architecture can also be used for lower cost alternatives such as the cost-reduced Thin Ethernet, an inexpensive Ethernet derivative that uses twisted pair wires to serve small work groups including those using standard IBM PC-software. Such lower-cost solutions will soon be widely available from Xerox and other vendors. The Xerox local area network is a single architectural solution with many physical realizations which include fiber optics and radio broadcasting. It has been adopted as an industry standard by the IEEE and the international standards organizations.

Internetworking

The internet transport protocols with their unique host address numbers and flexible range of protocols provide a networking solution, ideally suited to interconnecting many large distributed networks. With the use of these protocols, a user can connect to any system element on a world-wide network just as if it were a local system element. There is no need to use any central processors for communication control; the system automatically selects the best route for communication which adapts to changing network loads and configurations. This internet architecture, because of its unique features, has been adopted for use by dozens of other companies offering networking solutions.

Network services

The great variety of network services in XNS makes it versatile and useful. This includes support services (such as Clearinghouse, Time, and Authentication) as well as the application services (such as Print, File, Mail, and Gateway). The XNS services are general-purpose and provide compatibility to a wide array of products and services, including other vendor products. Flexibility and foresight has been built into every step of the architecture design to ensure reliable, smooth, and secure operation in a distributed environment that may span the globe.

Most other network architectures have not defined the protocols at the general applications level. The closest comparison in this area is IBM with its Document Interchange Architecture (DIA) which provides Document Library, Document Distribution, and Applications Processing services. These services are not as versatile, comprehensive, or general-purpose as XNS. Further, they do not work as well in a distributed environment.

XNS network services protocols are gaining wide recognition in the international standards organizations. For example, the Xerox Clearinghouse directory services model and the Mail Protocol model are being adopted by the CCITT as recognized international standards.

A particularly useful feature of XNS services is the Gateway Service which enables inter-operability with other network architectures, such as, IBM's SNA. The Gateway Service also enables terminal emulation capabilities so that XNS workstations can emulate IBM 3270, VT-100, or TTY terminals. The XNS emphasis on compatibility with other architectures and industry standards is crucial to users who wish to integrate diverse equipment and services to provide an integrated office system.

Character codes

No other Character Code Standard in the industry is as capable, as versatile, or as compatible with the many national and international standards, as the Xerox Character Code Standard. The Xerox standard can provide for a ***unique code for all of the languages in the world.*** This provides multilingual integration and avoids ambiguity. Further, it is efficient and compatible with ASCII, ISO, and the Japanese JIS Standards. IBM's EBCDIC lacks industry compatibility and is unable to represent the multinational character sets.

Document interchange

The Interscript document interchange standard is one of the most general-purpose and rigorous definitions available. Unlike other document representation standards, such as IBM's DCA Revisable Form, and ANSI's SGML (Standard General Markup Language), Interscript is suitable for WYSIWYG (what you see is what you get) editors and is fully code-independent. This representation is in a formative stage and is being evaluated by international standards groups and a number of major systems vendors.

Document printing

The Interpress Electronic Printing Standard provides a versatile, performance-oriented, page description language capability that makes it eminently suitable as an industry standard. Such alternatives as IBM's DCA Final Form and ANSI NAPLPS representations are too closely tied to specific hardware and lack the generalized document manipulation and font handling capabilities of Interpress. Other representations patterned after Interpress or its precursors, such as PostScript and Impress, lack many features important in an industry standard. PostScript, for example, has been designed to be suitable for creation by a person and does not maintain a clear separation between processes that belong in the creation domain and those that belong in the printing domain. This results in many design compromises which makes PostScript less suitable as a general-purpose electronic printing standard. PostScript may work well in a specific workstation printer combination where performance and uniformity are not important. However, PostScript lacks the page independence, the sequenceInsertFile, the compactness, the printing instructions, and many other capabilities of Interpress. Such capabilities are essential in a distributed environment where many different types of creators (and composition languages) would like to operate with many different types of output devices. Interpress has been adopted as a standard by many companies and is being considered by hundreds of others.

XNS features

Beyond the specific protocols and standards, XNS provides many features uniformly and consistently throughout the architecture. These include:

Reliability: By distributing and replicating key services so a system's operation is not compromised by the failure of a single machine.

Security: By protecting a system against unauthorized activities.

Flexibility: By accepting and supporting equipment made by others, and by allowing for distributed control.

Expandability: By making it easy to grow or change a network to meet user needs.

Compatibility: By enabling a variety of user applications with other architectures such as IBM's SNA.

Ease of Use: By providing the best ways for people to harness the power of the system.

XNS represents sophisticated network systems which are well integrated. Simpler systems that are not integrated but satisfy specific short-range objectives are always possible, often at a lower price. The real test of a system's quality, however, is if it can be used over a long period as applications change and new technology is introduced. In the long run, systems that lack proper architectural support are seldom the most cost effective. They become obsolete in a short time and have to be replaced, often resulting in costly disruption for users. Xerox Network Systems architecture, with its long-term perspective, is for those users and suppliers who want to do it right the first time.

Xerox is committed to the objective of compatibility among products from different vendors. It is working with other vendors as well as with standards organizations in making this possible. Xerox on its part will work to be compatible with, or to adopt, the important international standards as they emerge into the market.

These are some of the many reasons so many organizations are working with Xerox to solve their network architecture problems. They realize that Xerox Network Sytems architecture is right for today's problems and flexible enough to meet tomorrow's needs.

C. XNS AND INDUSTRY STANDARDS

Few technologies require more stable, comprehensive standards than communications. The whole point of communications technology is to permit parties to exchange information, and when those parties are using equipment designed by different companies, there is little hope that the exchange will be successful unless some degree of standardization has governed the design process.

There are two kinds of standards. *De facto* standards are set when one organization pioneers a particular technical approach, and when that organization is sufficiently prominent—due to technical innovation, market share, etc.—that its approach is accepted as "a good way to do it" by designers who come along later. Often, achieving compatibility with the products of the *de facto* standard-setter is an added motive. Examples of *de facto* standards are IBM's Binary Synchronous Communications (BSC) link control protocol and Xerox' Ethernet local area network.

Formal standards are set when a group of interested parties —usually drawn from industry—draw up a consensus specification for ratification by the industry community. To varying degrees, this process is assisted by various governmental scientific and economic agencies. Examples of formal standards are the RS-232-C communications interface specification and the ASCII code set.

Typically, formal standards are drawn up under the auspices of a standards-setting organization, which may or may not have official governmental status. Examples of such organizations include:

- American National Standards Institute (ANSI)
- Electronic Industries Association (EIA)
- International Telephone and Telegraph Consultative Committee (CCITT)
- International Organization for Standards (ISO)
- European Computer Manufacturers Association (ECMA)
- Institute of Electrical and Electronic Engineers (IEEE)

Users and suppliers alike have a stake in understanding the extent to which new systems are compatible with existing standards, and the extent to which the new systems will become compatible with relevant new standards. This applies also to areas where *de facto* standards have operated informally for a period. Here, the interest is to "legitimatize" the original *de* *facto* standards through a process of formal adoption, ensuring that the stability of the standard is not in the hands of a single organization.

With regard to its Xerox Network Systems architecture, Xerox has adopted the following position concerning industry standards:

In the design of Xerox Network Systems, and in the product implementations of the architecture, Xerox will be compatible with, or adopt, the important international standards as they emerge into the market.

Instances of this commitment can be found in a number of places in XNS including, for example:

- IEEE 802 Local Area Network standards: the work of this committee has been supported by Xerox from its inception. The baseband version of the 802.3 CSMA/CD Media Access standard, in particular, is the "formalization" of Xerox' Ethernet access method technique. Xerox has made nominal adjustments to its original Ethernet specification to be compatible with the approved version of the standard.

- Mail Standards: as CCITT X.400 series of electronic mail and message standards emerge into the market, Xerox anticipates that its own mail format standards will remain compatible, having been designed with the intent of compatibility in mind.

- Character Codes: Xerox has made a vigorous attempt to implement a set of internal codes for characters and character-like units of information that adhere to various U.S. and international standards. This attempt has been complicated by the fact that the existing standards are mutually inconsistent in a number of details. Xerox has adopted the ISO code standards as the basis for its own internal codes, and anticipates designing conversion logic in its products when those products are required to interface with external systems adhering to non-ISO standards.

- Transport Protocols: as ISO work on transport-layer protocols coalesces into formal status, Xerox anticipates designing gateway services that will permit its XNS transport-layer protocol, Internet, to function successfully with systems designed to the new ISO standard.

D. EXAMPLES OF XNS PROTOCOL USAGE

The following examples show the operation of XNS systems through the several common processes at each protocol layer.

Clearinghouse requests

In this example, the user needs to locate a system resource (such as a printer). Before doing that, however, the user must first locate a Clearinghouse server. Fig. D-1 shows the process in which a client causes an Ethernet broadcast message to be issued, which is responded to by the Clearinghouse server with its own address. Now the user can interrogate it for the name and address of resources available which match the desired profile. The process is similar for Authentication and Time Services.

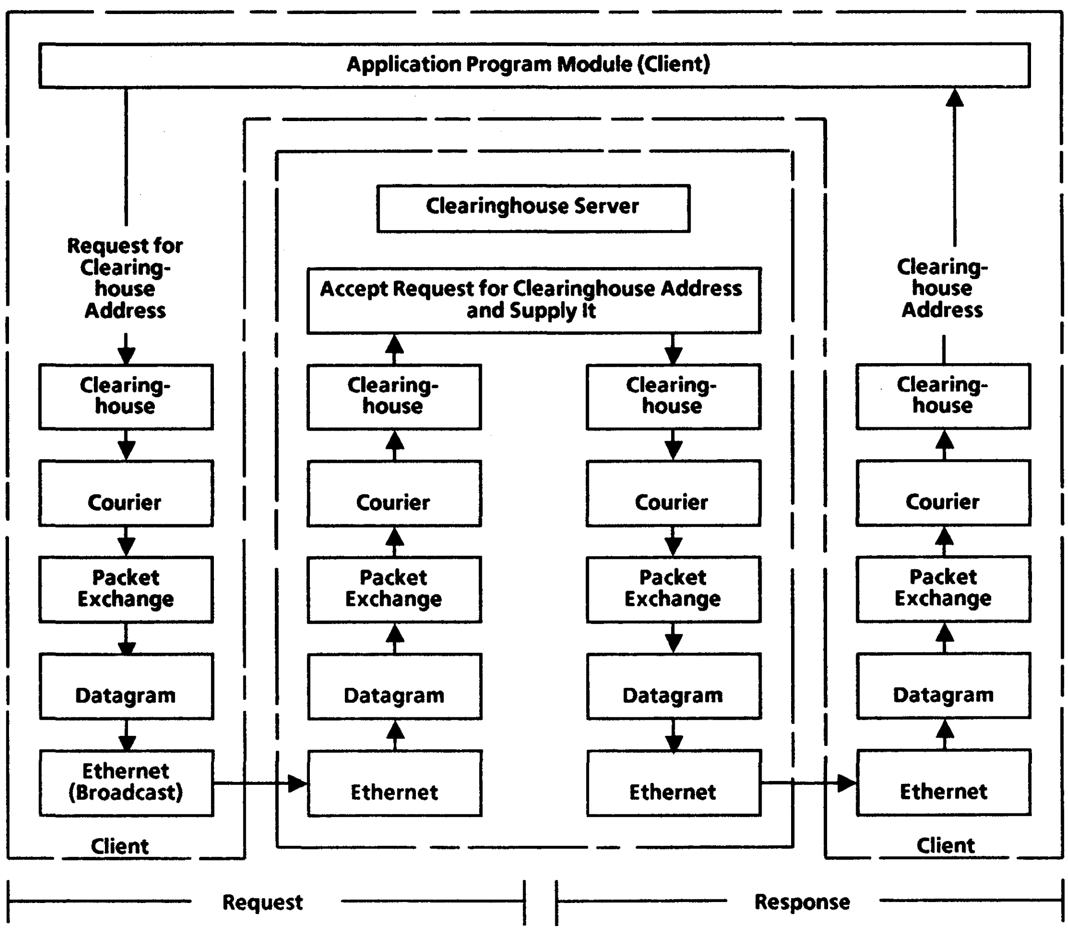


Figure D-1 **Clearinghouse request and response**

Remote printing

The user wishes to print a document on a printer located at a remote Ethernet location. In this case the Internetwork Router Service is used to pass information between two distant Ethernet networks, as shown in Fig. D-2. In the interest of simplifying the chart, only a single pass through the process is shown; in fact, the printer control information would be passed as conventional Courier exchanges and the actual data to be printed (encoded in Interpress form) would be passed by means of Bulk Data Transfer.

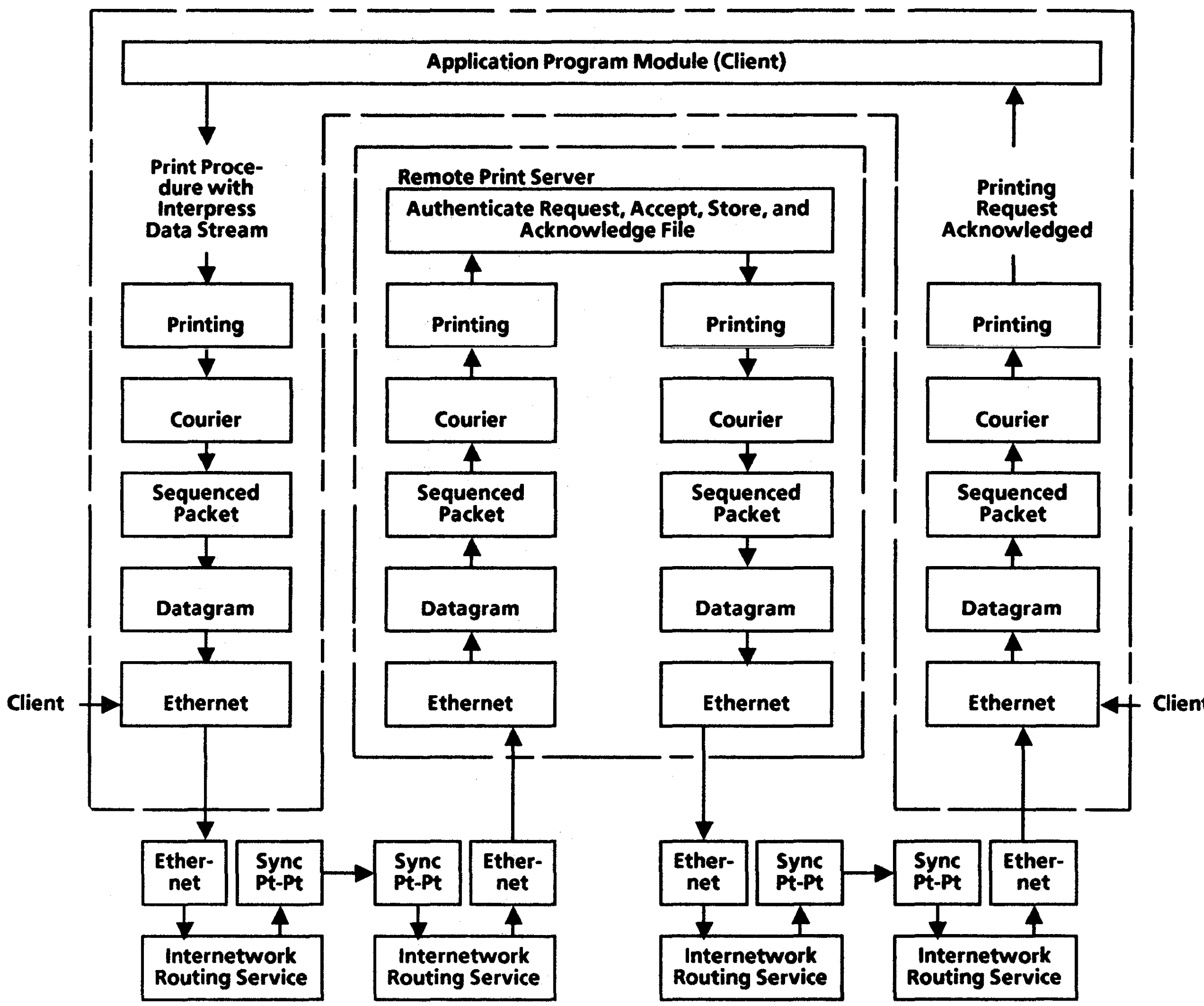


Figure D-2 **Remote printing**

Filing with authentication

In this example, the user wishes to file information at a file server. The file server first authenticates the request, then proceeds with filing and sending an acknowledgement to the user, as shown in Fig. D-3.

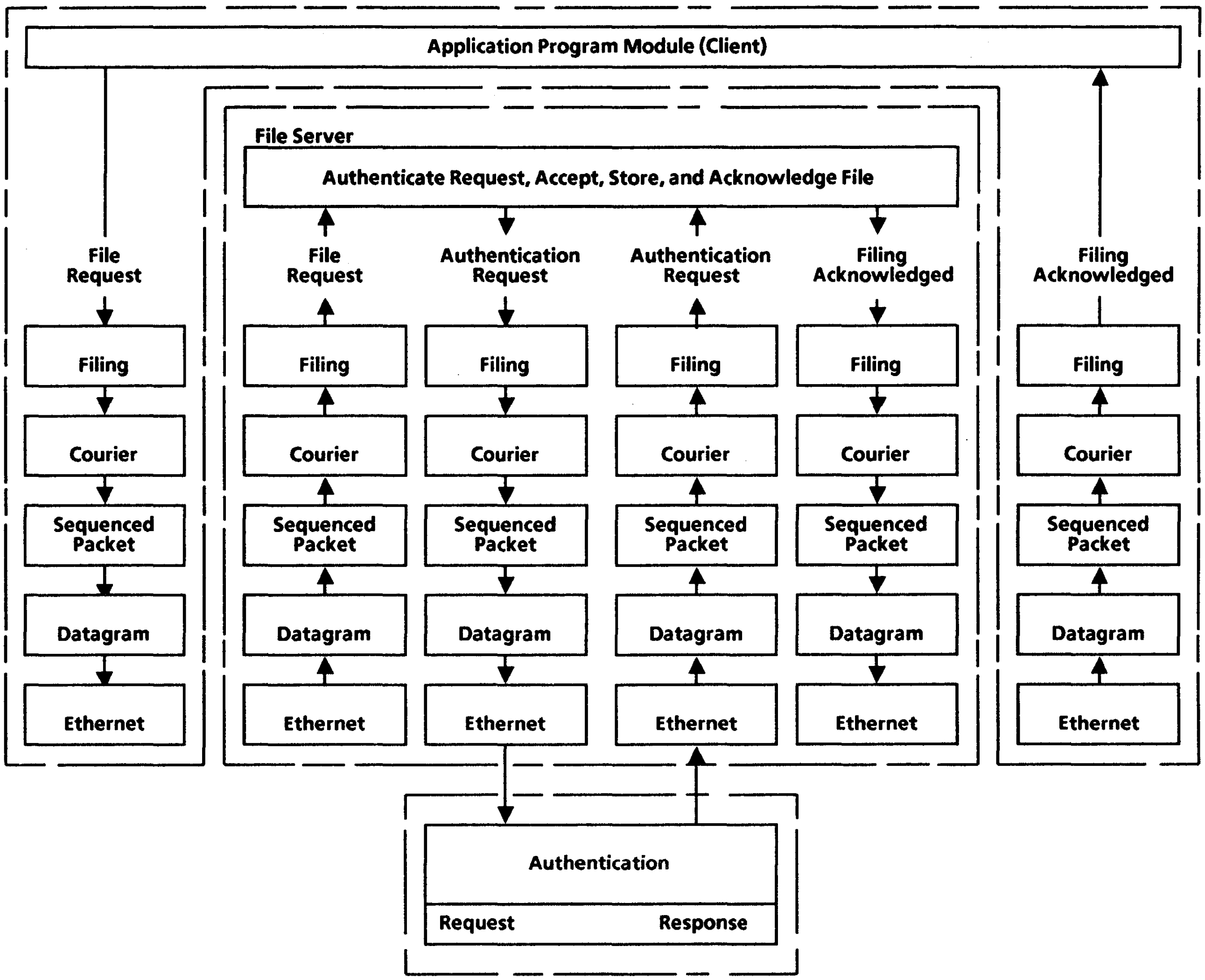


Figure D-3 **Filing with authentication**

Sending mail

The user wishes to send mail to a designated recipient. Mail Transport (and Bulk Data Transfer) is used to send the message, as shown in Fig. D-4. In operation this is similar to that modeled by CCITT in its X.400 series of proposed standards, except that in the XNS case the message is typically posted at a mail server, rather than directly with the recipient.

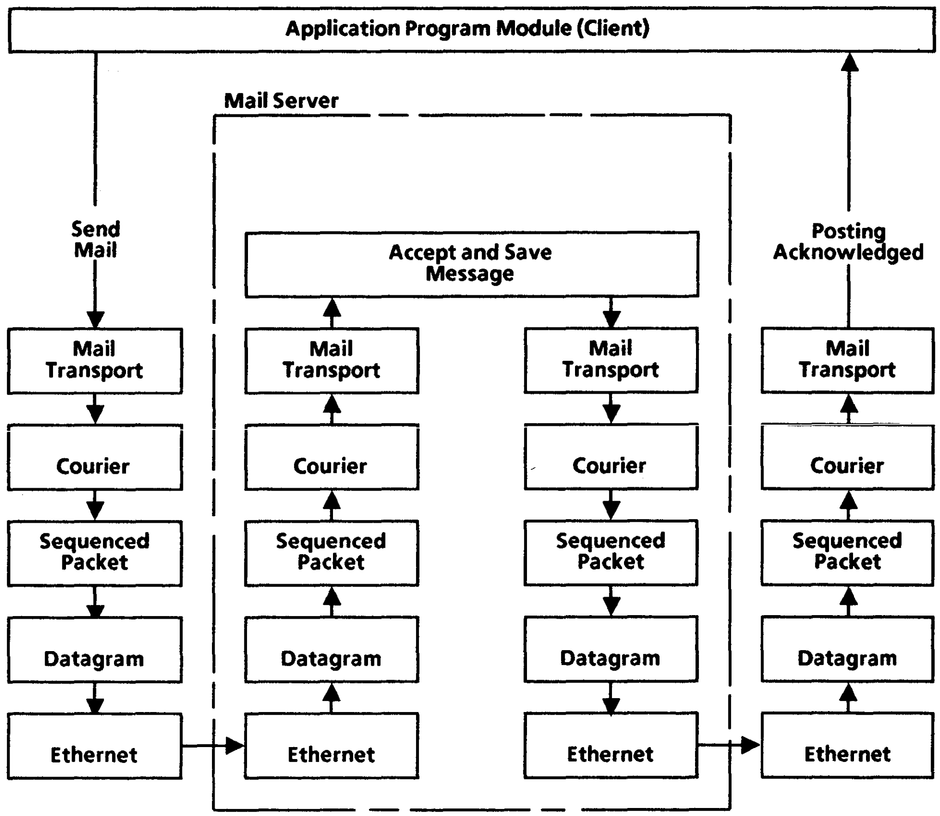


Figure D-4 **Sending mail**

Receiving mail

This is the other half of the transaction begun in Figure D-4. Here, the user is the recipient of the message, who uses the Inbasket Protocol to obtain messages from the mail server, as shown in Fig. D-5. As before, transfer of the message itself is by means of Mail Transport (and Bulk Data Transfer). An advantage of this approach is that the user can receive mail at more than one physical location.

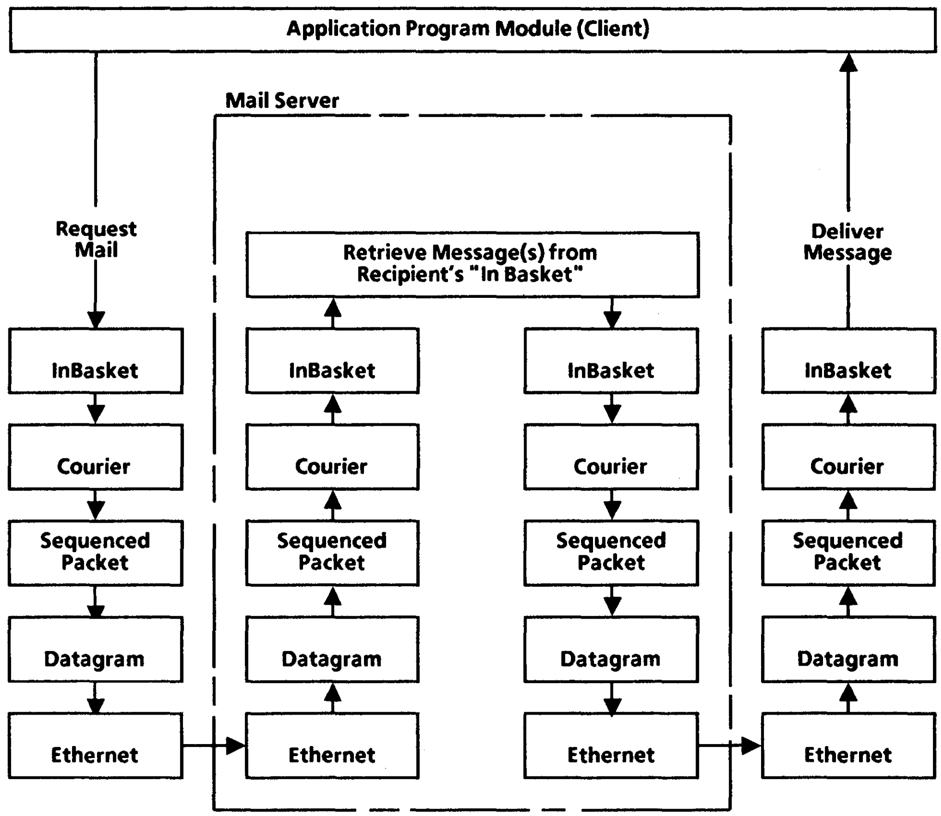


Figure D-5 **Receiving mail**

E. GLOSSARY

This glossary contains terms used in this manual. The terms are defined in the context of Xerox Network Systems (XNS) architecture. Italics within a definition indicate that the term is defined in this glossary.

Adaptive In *Raster Encoding Standard,* a Xerox standard compression technique which adapts to the image properties.

AI **Artificial intelligence** A computing concept in which combinations of hardware and software collect information in a manner that permits the drawing of inferences, hence "learning"; also called "knowledge-based systems."

ANSI **American National Standards Institute** This body specifies U.S. standards; such as, *ASCII.*

ASCII **American Standard Code for Information Exchange** A digital code set which represents each character of the standard typewriter keyboard as a 7-bit digital code.

Asynchronous Transmission of data in which time intervals between transmission can be unequal, controlled by start and stop elements at the beginning and end of each character.

Attributes In *Filing,* attributes are additional information about the file (e.g., file name, size, version, time created, access list, etc.)

Authentication The process which helps clients and services determine each other's identity in a reliable and secure way.

Baseband A local area network transmission technique in which the electrical signaling on the medium directly reproduces the digital form of the information; contrast with *broadband.*

Base language The syntax and semantic framework of Interpress or Interscript.

Bit synchronous A form of synchronous transmission in which single symbols (bits) are transmitted in a continuous stream.

Broadband A local area network transmission technique in which the digital signal is used to modulate a carrier signal, which is often combined with other carrier signals; contrast with *baseband.*

Broadcast The class of media for which the *Ethernet* is designed, in which all stations are capable of receiving a signal transmitted by any other station. Also, mode of usage of such a medium by the *Data Link Layer* in which all stations are instructed to receive a given frame.

BSC **Binary Synchronous Communications** A data link control procedure developed by IBM; includes specific control characters and procedures.

Bulk data In *Courier,* an arbitrarily long (usually quite long) sequence of 8-bit bytes, optionally interpreted as a single data object.

Bulk Data Transfer Protocol A protocol adaptation of *Courier,* for transfer of *bulk data.* Includes both an *immediate transfer* and a *third party transfer* mode.

Carrier sense Signal provided by the *Physical Layer* to the *Data Link Layer* of *Ethernet* to indicate that one or more stations are currently transmitting on the channel.

CCITT **Consultative Committee for International Telephone and Telegraph** An international standards group for communication standards. The *CCITT* standards recommendations usually have an "X" prefix, and are defined under "X" (e.g., X.25, X.400).

CCITT-4 In *Raster Encoding Standard,* a standard compression technique developed by *CCITT* for Group 4 facsimile devices.

CIU **Communication Interface Unit** An *RS-232-C port* extender for the *Gateway* Services; may be referred to as an 873.

Character A graphic shape that is used for representation of visual information. In *electronic printing,* a character is represented in the form of a spatial arrangement of adjacent pixels.

Character code Any code representing a *graphic character, a rendering character,* or a *control character,* usually unique within the set.

Character set An instance of character collection having a fixed number of characters.

Circuit A communication link between two points.

Clearinghouse An XNS service that provides a central repository for names, addresses and properties of system resources, permitting XNS system elements (workstations, servers, etc.) to locate them.

Client A person, a person and *workstation,* or a logical process that initiates some service, and that benefits from the service having been provided.

Clusternet A *network* formed by a cluster of XNS *workstations* connected via Routing Service, making them part of *internet*.

Clusternet Router A system which provides routing information to the *clusternet* and routes packets between clusternet circuits and the rest of the internetwork.

Collision detection A signal provided by the *Physical Layer* to the *Data Link Layer* of *Ethernet* to indicate that one or more other stations are contending with the local station's transmission. It can be true only during transmission.

Composition The process of organizing text and graphic information into a page form ("made-up pages"), as part of an overall publishing process.

Compressed In *Raster Encoding Standard*, a standard *compression* technique used within the *IMG* format.

Compression A computation that reduces the number of bits required to specify some data, usually a *pixel array*.

Control character A character, other than a graphic character or rendering character, whose occurrence in a particular context initiates, modifies or stops a control operation.

Courier An XNS protocol permitting the initiation and control of remote processes, including the transfer of information and control parameters associated with such processes.

CRC **Cyclic Redundancy Check** Used for error control in *Ethernet*.

Credentials An encrypted information set provided by the *Authentication* service to ensure security.

CSMA-CD **Carrier Sense Multiple Access with Collision Detection** The generic term for the class of link management procedure used by the *Ethernet*, which a) allows multiple stations to access the broadcast channel at will, b) avoids contention via *carrier sense* and difference, and c) resolves contention via *collision detection* and retransmission.

Datagram A unit of transmission in certain communication systems, over which individual routing and accounting control is exercised, and which is a constituent building block of a message.

Data Link Layer The higher of the two layers in the *Ethernet* design, which implements a medium-independent link level communication facility on top of the *Physical Layer*. (Also Layer 2 of the *ISO Model*.)

DCA **Document Content Architecture** IBM's architecture for the content of documents which include a revisable form as well as a final form specification.

Decompression Expansion of compressed data into its original form.

DES **Data Encryption Standard** from NBS (National Bureau of Standards).

Destination address The *host* number, *network* number and *socket* number of the destination in *internet*.

Device-independent A system which does not depend on properties of the device .

DIA **Document Interchange Architecture** IBM's architecture for the location, distribution, and processing of documents.

Directory In *Filing*, the organization of files into a structure.

Distributed system A system whose intelligent elements are interconnected by means of a communication system.

Distribution list A list of recipients for *electronic mail* represented by a user group registered in the Clearinghouse.

Document management Processes and facilities that permit creation, modification, and production of documents.

Domain Logical grouping, usually geographical, of registered objects in the Clearinghouse. It is also one part of the fully qualified three-part *Clearinghouse* name.

Drop cable The cable connecting a *transceiver* to a *network device*.

Duplex A mode of printing in which images are placed on both sides of a sheet of paper; also a *printing instruction* in Interpress.

EBCDIC **Extended Binary Coded Decimal Interchange Code** The dominant Character Code Standard for IBM computers.

ECMA **European Computer Manufacturer's Association** This group plays a vital role in establishing European standards.

ECS **External Communication Service** Service which provides terminal *emulation* services and controls the *RS-232-C* port of the *CIU*.

EIA **Electronic Industries Association** This group has established many electronic interface standards, including the RS 232-C.

Electronic mail A non-interactive communication between people that is transported electronically, not physically.

Electronic printing Use of a laser-based xerographic reproduction system to print encoded information in a wide variety of typefaces and styles, image orientation, graphic techniques, etc.

Electronic publishing Use of electronic techniques to produce finished hardcopy publications from text and graphic input.

Electronic reprographics A technique of reproducing hardcopy by electronically scanning it, and then printing copies of the scanned image on an electronic printer.

Emulation The process by which *workstations* behave like other types of *terminals*. Provides access to other *host* systems in a way familiar to users of those systems.

Encryption The technique for encoding information in such a way that it can only be deciphered by those with a key to decoding it.

Ethernet The Xerox *baseband* local area network technology.

External instructions In printing, those *printing instructions* that are supplied by mechanisms outside an Interpress master.

External Mail Gateway Option of the *Mail Service* that allows separate XNS *internets* to exchange mail

Facsimile **FAX** A system of communications in which a document, photograph, or other hardcopy graphic material is scanned, and the information transmitted to a remote receiver, where the image is reconstructed and duplicated onto paper.

File A set of related records treated as a unit.

File format The arrangement and structure of data or words in a file, including the order and size of the components of the file.

File Service Service which provides filing facilities to clients on the *internet*.

Font A particular collection of characters of a typeface with unique parameters, *i.e.,* one size and one face.

Font file A set of font records including a digital representation of a set or collection of graphic symbols and/or characters and control information.

Formatting The conversion of a document to a new desired form. In the *Printing Protocol* model, the second phase in which an *Interpress master* is converted to a form suitable for rendering by a specific printer marking engine.

Full-duplex A *circuit* or a *protocol* that permits transmission of a signal in two directions simultaneously.

GAP **Gateway Access Protocol** An *XNS* protocol that enables compatibility between *XNS* and non-*XNS* system elements.

Gateway Services that provide compatibility between systems of differing network or communication architectures.

GHO **Graphic Handling Option** An option in the Xerox 9700/8700 Electronic Printing System that enables the capability to handle *scanned images*.

GKS **Graphical Kernel System** An ISO graphics standard.

Graphic character A character, other than a *control character* or *rendering character*, that is normally represented by a graphic.

Half-duplex A *circuit* or a *protocol* that permits transmission of a signal in two directions, but not both directions at the same time.

Halftone The reproduction of continuous-tone artwork, such as a photograph, through a crossline or contact screen, which converts the image into dots of various sizes.

HDLC **High-Level Data Link Control** An ISO *Data Link Layer bit-synchronous* communication protocol for data links. Function similar to *SDLC* and *Synchronous Point-to-Point* protocol.

Heterogeneous system In local area networking, a communication system capable of supporting transmission among various groups of devices, not all of which are similar.

Hierarchical System A computer/communication system organized around a central processor, to which all other elements are subordinate in operation and data flow.

Homogeneous system In local area networking, a communication system in which all connected devices are compatible.

Host Any computer that is accessed by users. Commonly referred to as a mainframe as opposed to a workstation or service. (Sometimes used for XNS system elements.).

Icon In advanced display systems, a graphic element, such as a labeled box or other simple representation, that stands for a source, destination, or process.

IEEE **Institute of Electrical and Electronic Engineers** A professional organization that also defines standards, such as the *IEEE* 802.3 standard for local area networks (*LAN*).

Imaging model The process whereby primitive images specified by a color and a mask are built up on a page image.

IMG A format for representing a *scanned image*, used by the Xerox 9700. Now a part of the *Raster Encoding Standard.*

Immediate transfer A simple form of *bulk data transfer* in which the initiator is either the sender or the receiver .

Inbasket Protocol In *XNS* mail, the protocol for fetching messages received in a user's mail box at a *mail server.*

Information transcription The transfer of information from one physical system to another, often requiring reblocking.

Information translation The alteration of information contained in one format by expressing it in another format.

Integration The act of combining various system elements so that they can work productively together.

Intelligent scanning A technique of scanning which performs intelligent *character* recognition while scanning. Intelligent Character Recognition (ICR) is different from Optical Character Recognition (OCR) in that the system can be trained to recognize almost any type of character.

Internet (Also, **Internetwork**) A composite of interconnected networks in which all elements attached to any of the networks can communicate to each other as part of a unified system.

Interpress A Xerox standard for encoding the description of the final form of a page or document to be printed.

Interpress master A file written according to the *Interpress* Standard.

Interscript A Xerox standard for encoding a document in revisable form for purposes of exchanging the document between two or more dissimilar editors.

IRS **Internetwork Routing Service** The service that physically interconnects separate networks to form an internetwork, allowing the sharing of resources among networks.

ISO **International Organization for Standardization** An organization that sets international standards in many areas, including information systems.

ISO model The ISO Open Systems Interconnection Reference Model, a conceptual scheme in which the functions common to network architectures are arrayed in a layered hirerarchy. Widely known as the OSI model, it has also been adopted by the CCITT.

ITS **Interactive Terminal Service** Service which allows users of remote terminals and personal computers to perform mailing and filing functions.

Justify Spacing characters out so that they completely fill a predetermined region.

Kerning A technique for altering intercharacter spacing of pairs of characters to achieve a better, more compact visual appearance. (*Kern* is the portion of the character which extends beyond its body.)

LAN — **Local Area Network** A communication system intended to support exchanges of data and commands between stations within a confined area, such as a single building, a university campus, etc. Contrast with *wide area network.*

Layout structure — In *Interscript*, the way a document is to be rendered.

Ligature — A printed character consisting of two or more characters joined together as a single graphic.

Logical structure — In *Interscript*, the way a document is organized.

Logotype — (also, **logo**) A symbol, image, or complex character, composed of multiple entities which, when assembled as a unit, provide a given graphic shape.

Logoff — A term used for the process of disconnecting from a computer system, such as a file server.

Logon — A term used for the process of beginning a session with a computer system, such as a file server.

Mailbox — In *XNS* mail, the place where a user's mail is kept before it is retrieved by the user. Generally identified by a user's name and the name of the mail service where the user is registered.

Mail format — The Xerox standard for the content of a mail message (aligned with the CCITT X.420P2 specification).

Mail Service — Service providing mail facilities to clients on the network.

Mail Transport Protocol — In *XNS*, the protocol for posting messages to the mail system.

Marking — The actual creation of an image on paper. The third and final phase in the *Printing Protocol* model in which the document is actually printed.

Mask — In *Interpress*, the concept of a geometric shape that is laid over the page image in order to determine where to apply the colored ink to the image.

Modem — Device that converts digital information into an analog signal suitable for sending over analog phone lines; also converts the analog signal from phone lines into digital information.

Mouse — A pointing device used with advanced workstations to facilitate user interaction with screen content.

Multicast — An addressing mode in which a given frame is targeted to a group of logically-related stations.

NAPLPS — **North American Presentation-Level Protocol Syntax** ANSI standard for Videotex/Teletext transmission of text and graphics.

Network Physical and logical connection of system elements.

Network Address *Host* number, *network* number and *socket* number of an entity, constituting its address on the *internet*.

Network Architecture The overall design for a computer communication system, consisting of a structure and a set of protocols which precisely guide the implementation of the functions falling within the architecture's scope.

Network device A device connected directly to an *Ethernet*. Usually synonymous with *XNS* system element.

Open-endedness The quality of system design that permits future extensions and modifications with little, if any, backward impact.

Open system A computer communication system whose architecture and protocols have been rigorously defined, are under design control, and have been made available for external use or interface.

OSI Model Open Systems Interconnection Reference Model adopted by the ISO and the CCITT. (see ISO Model)

Packets A collection of data to be transmitted, typically containing routing and error-correction information.

Password A required code in addition to a user's name that allows logon, required for using a service, or for authentication.

Pathnames The means for specifying the route to objects that are nested within other objects.

Physical Layer The lower of the two layers of the *Ethernet* design, using the specified coaxial cable medium. The Physical Layer insulates the *Data Link Layer* from medium-dependent physical characteristics (also Layer 1 of the *ISO Model.*)

Pixel A picture element which is an element of a pixel array.

Pixel array A two-dimensional array of samples that define the color everywhere in a rectangular region.

Point size In reference to the size of a character, the normal spacing between lines of type of that size, e.g., 10-point type is sized so that it will be most legible when printed in lines spaced 10 points apart.

Print request identifier An identifier used by a *Client* to track the progess of a print job at a *print service*.

Printing instructions Commands that control the printing of an *Interpress master*, communicated within the master, or via the *Printing Protocol.*

Printing Protocol The *protocol* for printing documents on a *Print Service*.

Print Service A *service* that provides the resource for obtaining printed output of documents to *clients* accessing the *network*.

Priority Important In *Interpress*, the parameter used to designate when the printing sequence must match the Interpress presentation sequence.

Protocol A set of rules agreed to by two communicating parties for accomplishing some specific set of tasks.

Protocol layering A technique of structuring protocols so that the protocol at a layer uses the protocol at the next lower layer without knowing the details of its operation.

PSIS **Print Service Integration Standard** The standard defining integration with Xerox Print Services.

Raster A two-dimensional array of *pixels* that covers an image.

Raster scan Process of methodically scanning past each *pixel* on the image.

RBS **Remote Batch Service** Service which provides document interchange with devices and systems which implement IBM Binary Synchronous (*BSC*) data transmission protocols (*2770*, *2780* and *3780*).

Reference Subset A standard subset of *Interpress* adopted by Xerox Print Services as a minimum level capability.

RES **Raster Encoding Standard** An *XNS* standard for encoding *scanned images*.

Registry A dynamic and sometimes complex service whose purpose is to record information, assign and register unique identifiers, and respond to requests for information about identifiers.

Remote Procedure Call Protocol In *XNS*, same as *Courier.*

Rendering A version of a symbol produced by a process, such as printing.

Rendering character A character, other than a *control character* or *graphic character*, that fits into one of four classes: a non-conventional representation of a control character, a sequence of graphic characters (ligature or accented character), a contextually-dependent alternate representation for a graphic character (initial, medial, or final), or a "variant" representation for a graphic character.

Repeater A device used to extend the length and topology of the physical channel beyond that imposed by a single segment, up to the maximum allowable end-to-end channel length.

Replicative system A computer communication system capable of graceful, indefinite expansion through the incremental addition of new elements without requiring the modification or replacement of existing elements.

Router A logical process responsible for dispatching datagrams to their appropriate designations.

RS-232-C Port The physical interface between the server and the data communication equipment (usually a modem) which follows the EIS RS-232-C interface standard.

Scanned-image See pixel array.

Script In *Interscript*, the name given to the encoding of a document for document interchange purpose.

SDLC (Synchronous Data Link Control) An IBM communications line discipline or protocol associated with SNA.

Sequence Insert File In *Interpress*, the command that allows the specification of a file (such as a *Raster Image File*) within an *Interpress* master. The file itself may reside at a file system on the printer or anywhere on the *internet*. At the time of printing, the print service obtains and inserts the file in sequence.

Server A combination of hardware and software capable of performing some particular set of services.

Service A set of tasks performed on behalf of a client; a set of software resources which implement a number of XNS protocols to achieve a result on behalf of a client.

Session An association between a client and a system, by which the exchange of information is managed. (Used in the Gateway Access and Filing Protocols.)

Signature Normally a person's signature. In the printing industry, signature is the name given to a large printed sheet which folded (and cut) produces the final document (e.g., 2-up or 16-up signatures). In *RES*, the identification at the end of a file signifying that it conforms to *RES*.

Simple authentication A less secure form of *authentication*.

Simplex A mode of printing in which an image is placed on only one side of each sheet of paper; also a printing instruction.

SIU — **Shared Interface Unit** A unit which allows remote workstations to access the *Ethernet* and services by passing packets between the Ethernet and a phone line connected to the workstation.

SMS — **Server Monitor Service** Service that watches a group of servers from one location, monitors their availability, and reports problems to a System Administrator via mail messages.

SNA — **System Network Architecture** IBM's network architecture for mainframe/terminal communication.

Socket — An abstract location in an XNS system element that can originate or receive communication.

Source address — The *host* number, *network* number and *socket* number of the source in *internet*. Also called Source Network Address.

Spooling — The queuing of documents for printing. (The first phase for printing in the *printing protocol* model.)

Strong authentication — A very secure form of *authentication*.

Synchronous Point-to-Point Protocol — The Xerox *bit synchronous* communication protocol for point-to-point data links, in the context of *internet*.

System administrator — A user, with specific network privileges, who is responsible for setting up and maintaining the services and the organization of network services.

System element — Any computer on a network regardless of role. It includes workstations and servers. XNS system elements follow XNS *protocols*.

TCP/IP — The Arpanet Transmission Control and Internet Protocols.

Teletex — A European standard for text communication and electronic mail, introduced as an enhancement and eventually a replacement of Telex services.

Terminal — Generally, any device at an extremity of a communication system, capable of originating or receiving communication; in XNS any such device not directly connected to the local area network. Contrast with *workstation*.

Third Party Transfer — A form of *Bulk Data Transfer* in which the initiator is a third party, that is, it is neither the sender or the receiver.

Transceiver — The portion of the *Physical Layer* implementation that connects directly to the coaxial cable and provides both the electronics which send and receive the encoded signals on the cable, and the required electrical isolation.

Transmission medium — The lowest-level physicial transport mechanism, *e.g.*, leased lines, and *Ethernet*; also, a virtual transport mechanism.

Transport service A set of functions offered via an interface that provides transparent transfer of data between a client and a correspondent at the same level.

TTY A teleprinter terminal made by the Teletype Corporation. An *asynchronous* protocol used by such a device and most *asynchronous* terminals.

Typeface The features by which a character's design is recognized; hence the word "face."

Typography The art of designing and placing letter forms to create a legible and pleasing effect.

Utility Programs In *Interpress*, programs provided to perfom master manipulations such as creation of 2-up *signatures*.

Virtual circuit A logical path through various elements in a communication system, established for purposes of conducting some specific exchange between two or more parties.

Virtual memory A mechanism (hardware and software) that provides the illusion of a large memory by combining a small memory with a large disk. This technique permits the user to treat secondary storage as an extension of main memory, thus giving the virtual appearance of a larger main memory.

Virtual terminal circuits A mechanism which allows interactions to occur between various network citizens for many applications that require asynchronous *ASCII* communication.

VT-100 A popular type of display *terminal*. The protocol used by such a *terminal*.

VTP **Virtual Terminal Protocol** (see *GAP*)

Wide area network A communications network for use over long distances. Contrast with *local area network*.

Width Term referring to a *character*; the spacing from one *character* to the next.

Window In an advanced display system a portion of the display set aside for a specific purpose or application, and cut away from, or overlaid on top of, the balance of the display content.

Workstation Any XNS system element intended for direct human interaction.

WYSIWIG **What You See Is What You Get** This is the technique pioneered by Xerox for presenting information on a display in the same form as it will appear when finally printed.

XDE — **Xerox Development Environment** A set of powerful and sophisticated software development tools which use the resources of XNS.

Xerox Character Code — The Character Code Standard adopted by Xerox which assigns a unique code for most of the languages of the world, and is compatible with many national and international standards.

XModem — An asynchronous communication protocol used to transmit files between devices such as personal computers and main-frames, and between personal computers and *ITS*.

XNS — **Xerox Network Systems** The combination of hardware and software that unites specialized devices into a network where the capabilities of a variety of workstations are enhanced by distributed services, in accordance with the Network Systems architecture specified by Xerox. Also, XNS architecture.

XNSI — **Xerox Network Systems Institute** The Institute provides a variety of services including documentation, consulting, and implementation aids to those interested in the XNS architecture. Members get special discounts, advance information, and other benefits for a fee.

XNSIG — **Xerox Network Systems Implementors Group** This is a group of organizations who have implemented or are interested in implementing the XNS architecture. The Group meets several times a year to work on solutions to common problems.

X.25 circuits — An international standard protocol from *CCITT* for accessing packet-switching networks. The networks support virtual circuits between network access points.

X.400 series — These series of standards from *CCITT* pertain to electronic message handling systems (mail).

850/860 Gateway — Allows standalone 850 and 860 Information Processing Systems to exchange information (using the Mail Service) with other workstation users on the internet.

2770, 2780, 3780 — The family of IBM Remote Batch Terminals with which the Remote Batch Service communicates.

3270 Communication Protocol — Optional software running with *ECS* that supports compatible information exchange between workstations, with 3270 emulation software and an IBM host.

XNS standards and guides

This section contains references to the XNS standards and guides that comprise the specification of the XNS architecture. References to other standards which are part of the architecture or strongly supported within XNS are also included. The documents are listed under the section name in the order in which each section appears in this document.

1. Introduction

Introduction to Xerox Network Systems, XNSG 058504. April 1985.

This guide provides a non-technical overview of the Xerox Network Systems. It serves as an introduction to the more detailed discussion of the XNS architecture provided in this General Information Manual.

2. XNS Overview

ISO Open Systems Interconnection—Basic Reference Model. International Organization for Standardization. ISO/TC 97/ SC 16 N 719. August 1981.

Marked-up Reference Model of OSI (DP 74 98) to Add Connectionless Data Transmission. ISO/TC 97/SC 16 N 741. September 1981.

The above two documents define the ISO OSI reference model.

Xerox Network Systems Architecture General Information Manual, XNSG 068504. April 1985.

This guide provides a technical description of the XNS architecture. It is a road map to the standards and protocols that comprise the architecture and includes descriptions of XNS services and an annotated bibliography.

3. Communications

Data Communications-High-Level Data Link Control Procedures—Frame Structure. ISO International Standard (IS) 3309.

This reference defines the HDLC framing conventions.

EIA RS-232 Standard: Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange. Electronic Industries Association. August 1969.

This reference defines the RS-232-C physical layer interface and protocol.

The Ethernet: A Local Area Network (Data Link Layer and Physical Layer Specifications), Version 2.0. November 1982. Digital Equipment Corporation, Intel Corporation, Xerox Corporation.

This Version 2.2 of the Ethernet specification is a precursor to the IEEE 802.3 standard referenced below. The document contains the data link and physical layer specifications for the Ethernet.

IBM Synchronous Data Link Control. International Business Machines, Document No. GA27-30093-2.

This reference defines IBM's SDLC protocol.

Internet Transport Protocols, XSIS 028112. Xerox Corporation. December 1981.

This standard describes an open-ended set of internetwork packet transport protocols used uniformly across a variety of communication media, digital processors and office applications.

Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Standard 802.3-1985. [Available from IEEE.)

This edition of the standard defines a 10 Mb/s baseband implementation of the Physical Layer using the CSMA/CD access method.

Recommentation X.25, Interface between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks. Consultative Committee, International Telephone and Telegraph [C.C.I.T.T.]. Geneva: International Telecommunications Union. v. 7.

Synchronous Point-to-Point Protocol, XSIS 158412. Xerox Corporation. December 1984.

This document describes the protocol for the interconnection of individual communicating entities or data terminal equipment over a data link.

4. Remote Procedures: Courier

Courier: The Remote Procedure Call Protocol, XSIS 038112. Xerox Corporation. December 1981; with
Appendix F: Bulk Data Transfer, XSIS 038112 Add. 1. Xerox Corporation. October 1982.

This protocol defines a single request/reply or transaction discipline for an open-ended set of higher-level application protocols. Appendix F contains a detailed description of how Third Party and Immediate Bulk Data Transfer operates.

5. Application Support Environment

7-Bit Coded Character Set for Information Processing Interchange. International Organization for Standardization. ISO 646-1973 (E).

This document defines a limited character set for information interchange. It is almost compatible with ASCII.

American National Standard Code for Information Interchange. American National Standards Institute ANSI X3.4-1977.

This reference specifies the United States standard character code set generally known as ASCII. [U.S. version of ISO 646]

Authentication Protocol, XSIS 098404. Xerox Corporation. April 1984.

This document defines a protocol which provides a method of protecting user passwords, and of verifying a user's identity.

Character Code Standard, XSIS 058404. Xerox Corporation. April 1984.

This is the Xerox 16-bit character code standard which also describes a byte-oriented encoding scheme. Includes codes for rendering character and provides multilingual capability.

Clearinghouse Protocol, XSIS 078404. Xerox Corporation. April 1984; with

Clearinghouse Entry Formats, XSIS 168404. Xerox Corporation. April 1984.

The Clearinghouse Protocol describes the Clearinghouse facility for locating named resources in terms of its Courier remote procedures and error responses. Clearinghouse Entry Formats defines certain Clearinghouse Directory Service property types and the structure of their associated entries.

Code of the Japanese Graphic Character Set for Information Interchange. Japanese Industrial Standard, JIS C 6226-1978.

Code Extension Techniques for Use with the Code for Information Interchange. Japanese Industrial Standard, JIS C 6228-1975.

The above two references specify the Japanese JIS character code standards.

The Data Encryption Standard. National Bureau of Standards, Federal Information Processing Standards Publication (FIPS PUB) 46. January 1977. National Technical Information Service, Springfield, VA.

This reference defines the basic DES encryption algorithm.

DES Modes of Operation. National Bureau of Standards. Federal Information Processing Standards Publication (FIPS PUB) 81. 1980. National Technical Information Service, Springfield, VA.

This reference discusses the various modes of using the DES encryption algorithm.

Font File Format Standard. [in preparation]

This document will provide a standard font file format for interchange purposes. The format will convey both fonts and/or font metrics to font-using devices.

Guidelines for Implementing and Using the NBS Data Encryption Standard. National Bureau of Standards, Federal Information Processing Standards Publication (FIPS PUB) 74. April 1981. National Technical Information Service , Springfield VA.

This reference provides guidelines for using the DES encryption algorithm.

Time Protocol, XSIS 088404. Xerox Corporation. April 1984.

The document describes the techniques and data formats for requesting time from the network time service, and the returned response.

6. Gateway Access

Gateway Access Protocol Specification. [in preparation]

This document describes a method of providing remote access to the transport service supporting communication with non-XNS systems.

7. Filing

Filing Protocol, XSIS 108210. Xerox Corporation. October 1982; **Appendix E Printer Subset of the Filing Protocol,** XSIS 108210, Rev. 1. Xerox Corporation. December 1984.

This document defines a protocol for interaction between clients and file services. The appendix describes a permissible subset of the Filing Protocol for use in print servers which implement a primitive file server capability.

Filing Protocol Extension. [in preparation]

8. Document Interchange

Interscript Standard. [in preparation]

This standard provides a means of representing editable documents that is independent of any particular editor and can be used to interchange documents among editors.

Introduction to Interscript. [in preparation]

This supplements the basic Interscript Standard and seeks to help a reader understand the underlying principles of Interscript and their application.

9. Mailing

CCITT X.400 Series Recommendations on Message Handling Systems. (Approved at Malaga - Torri Molinos), October 1984.

The X.400 series from CCITT contains the following eight Recommendations:

- X.400 - System Model-- Service Elements
- X.401- Basic Service Elements and Optional User Facilities
- X.408- Encoded Information Type Conversion Rules
- X.409- Presentation Transfer Syntax and Notation
- X.410- Remote Operations and Reliable Transfer Server
- X.411- Message Transfer Layer
- X.420- Interpersonal Messaging User Agent Layer
- X.430- Access Protocol for Teletex Terminals

NS Mailing Protocols. [in preparation]

These protocols specify a carefully layered mailing architecture that promotes compatibility while also allowing flexibility through transparent multi-level encapsulation.

10. Printing

Interpress Electronic Printing Standard, Version 2.1, XSIS 048404. April 1984.

This is the reference document for the Interpress Standard, Version 2.1. It is written with precision and accuracy in mind, and as a result is not very descriptive. This document is the authoritative definition of Interpress.

Interpress Electronic Printing Standard, Version 3.0. [in preparation]

This document includes additional commands for curves, arcs, and other advanced graphics not in Interpress 2.1. It will include all of the proven capabilities of the Research version of extended Interpress.

Interpress 82 Readers Guide, XSIG 018404. Xerox Corporation. April 1984.

A narrative companion to an earlier version of Interpress. Much, but not all, of the discussion applies to the current version.

Introduction to Interpress, XSIG 038404. Xerox Corporation. April 1984.

This supplements the basic Interpress standard and is intended to help users understand the underlying principles of Interpress and their application.

Print Service Integration Standard, XSIS 198412. Xerox Corporation. December 1984.

This Standard supplies interfacing specifications required to provide full compatibility between document creation devices and Xerox electronic printing services.

Printing Protocol, XSIS 118404. Xerox Corporation. April 1984.

This document defines the protocol used for communicating print requests from clients to print services, including the remote procedures and error responses in Courier terms.

11. Scanning

Raster Encoding Standard, XSIS 178412. Xerox Corporation. December 1984; with
Appendix B. 9700 Graphic Image Specification.

This standard, also known as "RES," defines the technique by which scanned images are digitally encoded. Appendix B specifies the IMG format used by the Xerox 9700 and 150 GIS.

Copies of the XNS standards and guides may be obtained from:

Xerox Corporation
Xerox Network Systems Institute
2300 Geng Road
Palo Alto, Ca 94303
(415) 496-6088

Technical papers and reports

This section contains references to technical papers and reports which discuss the underlying philosophy and rationale for the XNS architecture and its implementation. The documents are listed under the section name in the order in which each section appears in this document.

1. Introduction

Dalal, Y. K. **The Information Outlet: A New Tool for Office Organization.** Palo Alto: Xerox Corporation, Office Products Division. 1981. OPD-T8104. [A version of this paper appeared in *Proceedings of the Online Conference on Local Networks and Distributed Office Systems,* May 1981.]

This paper describes the way distributed offfice information systems permit an organization to control its conversion to the "office of the future" by permitting the system to evolve according to the needs and structure of the organization.

2. XNS Overview

Becker, Joseph D. **Typing Chinese, Japanese, and Korean.** *Computer.* January 1985. pp. 27-34.

Becker, Joseph D. **"User-Friendly" Design for Japanese Typing**. Palo Alto: Xerox Corporation, Office Systems Division.1983. OSD-T8301.

This paper describes the design of a "user-friendly" Japanese typing software system. The application of the same design to the typing of the Chinese language is also discussed.

Becker, Joseph D. **Multilingual Word Processing.** *Scientific American.* July 1984. pp. 96-107

This article describes the computer processing of Japanese, Arabic, and other non-English writing systems, particularly the problem of mixing languages rendered from left-to-right with those rendered from right-to-left.

Lipkie, Daniel E., Evans, Steven R., Newlin, John K., and Weissman, Robert L. **Star Graphics: An Object-oriented Implementation.** *Computer Graphics.* July 1982. 16(3): pp. 115-124.

This paper describes the design and implementation of Star graphics using an object-oriented technique.

Smith, D. C., Harslem, E., Irby, C., Kimball, R. **The Star User Interface: An Overview.** *Proceedings of the National Computer Conference.* June 7-10, 1982. Houston. pp. 515-528.

This paper describes the features of the Star system without justifying them in detail.

Smith, D. C., Irby, C., Kimball, R., Verplank, B., Harslem, E. **Designing the Star User Interface.** *Byte.* April 1982. 7(4): pp. 242-282.

This paper describes the principles behind the features in Star.

Xerox's Star. *Seybold Report.* 10(16), April 17, 1981; and
The Xerox Star: A Professional Workstation. *Seybold Report on Word Processing,* 4(5). May 1981.

These articles provide a review of the Star professional workstation.

3. Communications

Boggs, David R., Shoch, J. F., Taft, E. A., and Metcalfe, R. M. **Pup: An Internetwork Architecture.** Palo Alto: Xerox Corporation, Palo Alto Research Center. 1979. CSL-79-10. [Also appeared in *IEEE Transations on Communications.* April 1980. com-28:4, pp. 612-624.]

This report discusses the important design issues of internetwork architectures. It describes the simplicity of the "Pup" architecture and its fundamental end-to-end datagram approach.

Dalal, Y. K. **Use of Multiple Networks in the Xerox Network System.** *IEEE Computer Magazine.* October 1982. 15(10): pp. 82-92.

This article describes the major features of the XNS internet communication system, in particular its ability to use different kinds of networks.

Dalal, Y. K., and Printis, R. S. **48-bit Absolute Internet and Host Numbers.** *Proceedings of the 7th Data Communications Symposium.* October 1981. pp. 240-245. [Also published by Xerox Corporation, Office Products Division. September 1981. OPD-T8101.]

This paper describes how the Xerox host numbering scheme was designed in the context of an overall internet and distributed systems architecture.

The Ethernet Local Network: Three Reports. Palo Alto: Xerox Corporation, Palo Alto Research Center. 1980. CSL-80-2.

Includes **Ethernet: Distributed Packet Switching for Local Computer Networks,** by Robert M. Metcalfe and David R. Boggs. [Also appeared in *Communications of the ACM.* July 1976. 19(7).]

Practical Considerations in Ethernet Local Network Design, by Ronald C. Crane and Edward A. Taft

Measured Performance of an Ethernet Local Network by John F. Shoch and John A. Hupp.

The first paper introduces the Ethernet-style network, a multi-access broadcast packet-switched communication system, and presents the theory (CSMA/CD) that underlies it. The second paper discusses the design and implementation of the prototype Ethernet system, and the final paper presents performance results based on several years' practical experience with that system.

Postel, J., ed. **DoD Standard Internet Protocol.** *ACM Computer Communication Review.* 10(4): 2-51. October 1980. [Available from NTIS, Springfield, VA. ADA079730.]

Postel, J., ed. **DoD Standard Transmission Control Protocol.** *ACM Computer Communication Review.* 10(4): 52-132. October 1980. [Available from NTIS, Springfield, VA; ADA082609.]

The above two references describe the ARPANET TCP/IP protocols.

Shoch, John F. **An Annotated Bibliography on Local Computer Networks**. Palo Alto: Xerox Corporation, Palo Alto Research Center. 1979. SSL-79-5. [Also available as IFIP WG 6.4, Working Paper 79-1.]

This bibliography contains a wide range of material related to the general area of local computer networking. It is divided into two sections, one on local networks and one on the related subject of radio-based systems.

Shoch, J. F. **Internetwork Naming Addressing and Routing.** *17th IEEE Computer Society International Conference (Compcon).* September 1978.

Shoch, John, and Stewart, Larry. **Internetwork Experiments with the Bay Area Packet Radio Network.** Palo Alto: Xerox Corporation, Palo Alto Research Center. 1979. SSL-79-4.

This report describes the integration of the Bay Area Packet Radio Network into an existing architecture for network interconnection.

Shoch, J. F., Dalal, Y. K., Crane, R. C., and Redell, D. D. **Evolution of the Ethernet Local Computer Network**. Palo Alto: Xerox Corporation, Office Products Division. September 1981. OPD-T8102. [Also appeared in *IEEE Computer Magazine*. August 1981. 15(8): pp. 10-27.]

This report highlights a number of important considerations that affected the design of the Ethernet local computer network.

White, J. E., and Dalal, Y. K. **Higher-level Protocols Enhance Ethernet**. *Electronic Design*. April 1981. 30(8): pp. 33-41.

This paper describes the additional levels of protocol needed to allow communication between networks and between processes within different pieces of equipment from different manufacturers.

4. Remote Procedures: Courier

Birrell, Andrew D., and Nelson, Bruce J. **Implementing Remote Procedure Calls.** Palo Alto: Xerox Corporation, Palo Alto Research Center. 1983. CSL-83-7. [Also appeared in *ACM Transactions on Computer Systems*. February1984. 2 (1).]

This paper describes a package providing a remote procedure call facility.

Nelson, Bruce Jay. **Remote Procedure Call**. Palo Alto: Xerox Corporation, Palo Alto Research Center. 1981. CSL-81-9.

This report explores the essential properties of the remote procedure call, and describes the detailed design of an RPC mechanism.

5. Application Support Environment

Abraham, S. M., and Dalal, Y. K. **Techniques for Decentralized Management of Distributed Systems.** *20th IEEE Computer Society International Conference (Compson).* February1980, pp. 430-436.

Gifford, David K. **Cryptographic Sealing for Information Secrecy and Authentication.** Palo Alto: Xerox Corporation, Palo Alto Research Center. 1982. CSL-82-1. [Also appeared in *Communications of the ACM*. April1982. 25 (4).]

This paper describes a protection mechanism that provides general primitives for protection and authentication.

Israel, Jay E., and Linden, Theodore A. **Authentication in Star and Network Systems.** Palo Alto: Xerox Corporation, Office Systems Division. 1982. OSD-T8201.

This paper describes the approach used in Xerox' Star and Network Systems products for user authentication.

Marzullo, Keith A. **Maintaining the Time in a Distributed System.** Palo Alto: Xerox Corporation. Office Systems Division. 1984. OSD-T8401.

This report develops, analyzes and experiments with algorithms that implement decentralized, self-correcting time services.

Needham, R. M., and Schroeder, M. D. **Using Encryption for Authentication in Large Networks of Computer.** *Communications of the ACM,* 21:12. December 1978. pp. 993-999.

This paper discusses some of the principles of the Xerox authentication approach.

Oppen, D. C., and Dalal, Y. K. **The Clearinghouse: A Decentralized Agent for Locating Named Objects in a Distributed Environment.** *ACM Transactions on Office Information Systems.* July 1983. 1(3): pp. 230-253.

This paper discusses the Clearinghouse, a decentralized agent for supporting the naming of objects in a distributed environment.

8. Document Interchange

Gencode and the Standard Generalized Markup Language (SGML). GCA Standard 101-1983. Graphic Communication Association, Arlington, VA.

This describes the Document Markup Metalanguage.

Purvy, R., Farrell, J., and Klose P. **The Design of Star's Records Processing.** *ACM's Transactions on Office Information Systems.* First quarter, 1983.

This paper describes integrating traditional data processing functionality into Star, using standard Star documents for data definition, entry, display, update and report generation.

Roberts, T. L., and Moran, T. P. **Evaluation of Text Editors.** *Proceedings of the Conference on Human Factors in Computer Systems.* March 1982 15-17. Gaithersburg, MD. pp. 136-141.

This paper presents a methodology for evaluating computer text editors from the viewpoint of their users—from novices to dedicated experts.

9. Mailing

Birrell, A. D., Levin, R., Needham, R. M., and Schroeder, M. D. **Grapevine: Two papers and a Report.** Palo Alto: Xerox Corporation, Palo Alto Research Center. 1983. CSL-83-12.

Includes **Grapevine: An Exercise in Distributed Computing,** by Andrew D. Birrell, Roy Levin, Roger M. Needham, and Michael D. Schroeder. [Appeared in *Communications of the ACM.* April 1982. 25 (4).]

Experience with Grapevine: The Growth of a Distributed System, by Michael D. Schroeder, Andrew D. Birrell, and Roger M. Needham. [Appeared in ACM's *Transactions on Computer Systems.* February 1984. 2 (1).]

The Grapevine Interface, by Andrew D. Birrell.

These papers describe Grapevine, a distributed mail transport system on the Xerox research internet, report on experience with its use, and describe in detail the services Grapevine provides.

Redell, David D., and White, James E. **Interconnecting Electronic Mail Systems.** *Computer.* September 1983. pp. 55-63.

This paper describes standard interfaces for mail gateways.

10. Printing

Aesthetics vs. Technology: typography for electronic printers and video displays. *Seybold Report.* 11(11), February 8, 1982, and 11(12) February 22, 1981.

This is an excellent article by Chuck Bigelow on letterform design and typography for raster devices.

Graphical Kernel System (GKS)—Functional Description. International Organization for Standardization. ISO/DP 7942.

Describes the proposed GKS graphics package standard.

Hunter, R., and Robinson, A. H. **International Digital Facsimile Coding Standards.** *Proceedings of the IEEE.* July 1980. 68(79): pp. 854-867.

A good survey of facsimile formats.

Newman, W. M., and Sproull, R. F. **Principles of Interactive Computer Graphics,** 2nd edition. McGraw-Hill. 1979.

This book is an introduction to computer graphics, geometric representations and transformations, and raster graphics.

Proposed American National Standard for Videotex/Teletext Presentation Level Protocol Syntax. American National Standards Institute. ANSI X3-110-198x.

Draft of "North American Presentation Level Protocol Syntax." (NAPLPS standard)

Copies of the XNS standards and guides may be obtained from:

Xerox Corporation
Xerox Network Systems Institute
2300 Geng Road
Palo Alto, Ca 94303

Attention: Dennis Frahmann
(415) 496-6088

XEROX

Xerox Corporation
101 Continental Boulevard
El Segundo, California 90245

Printed in U.S.A. 4/85 (15M) 610P50768