1990

Internetworking: an analysis and proposal

Gary M. Diana

Internetworking: An Analysis and Proposal

Internetworking: An Analysis and Proposal

Gary M. Diana
April 26, 1990

Rochester Institute of Technology
Department of Computer Science

A Thesis, submitted to the Faculty of the
School of Computer Science and Information Technology,
in partial fulfillment for the Degree of
Master of Science in Computer Science

Approved by:

5/1/90

Associate Professor Dr. Andrew T. Kitchen
Chairman, Thesis Committee

5/4/90

Associate Professor Dr. James E. Heliotis
Reading Member, Thesis Committee

5/1/90

Associate Professor Henry A. Etlinger
Reading Member, Thesis Committee

Title of Thesis: Internetworking: An Analysis and Proposal

Date: May 22, 1990

Acknowledgements

This thesis represents the final efforts on my way to obtaining the degree of Master of Science in Computer Science. There are many people who supported me over the six years it has taken, and I would like to take this opportunity to recognize them.

I would like to first thank Dr. Ronald Sarner of the SUNY Institute of Technology and Dr. Leonard Smith of the SUNY College of Environmental Science and Forestry for their letters of recommendation into the Master's program. They were instrumental to my admission. I would also like to thank Eastman Kodak Company, specifically the Systems Solution Group within Business Imaging Systems. They not only provided release time and tuition support, but also a resource-rich environment in which to carry out my work. Rich Bartl, Dan Himes, and Mary Alice Flagler, of the Eastman Kodak Technical Library, assisted greatly in the procurement of research materials. They saved me many hours time.

I would like to thank my thesis committee for their assistance. Dr. Andrew T.Kitchen, Dr. James E. Heliotis, and Henry A. Etlinger provided technical, professional, and moral support. They made what could have been a long drawn-out process an enjoyable experience. They were especially patient with the early revisions of my work.

I would like to thank my family and friends for their support. They all listened patiently to complaints of too much work to do, programs that didn't work, and due dates. My parents, Louis and Marilyn Diana, made numerous trips to Rochester to babysit, while I did school work and my wife did her nursing work. I simply would not have been able to complete the coursework without their help. I would also like to thank David and Jim, for their interest and support in this endeavor.

Finally, I would like to thank my wife Sheryl and our son Gary Jr. While I was taking classes and during my long hours in the grad lab, she managed a career, a house, and a type A toddler. There is no one else who would have done this for me. Gary Jr. was very patient with a Dad who didn't always let him play with his computer, or have the time to play all the games three year olds like to play. I can now look forward to tee ball, fishing, roller skating, and bicycling with him... and rightly so.

For Sheryl, Gary Jr., and ???

Abstract

As the number of computer networks has grown, so has the desire for users on these networks to communicate with each other, thus the need for internetworking. Unfortunately, many of these networks were not designed with internetworking capabilities in mind. The internetworking facilities offered by a typical network range from non-existent to state of the art. Two major efforts towards internetworking are the DARPA Internet protocols and the OSI Internetworking protocols. The goals of this thesis are to acquaint the reader with the qualities which are desired in an internetworking scheme, to describe how internetworking is accomplished currently, and how these protocols might be modified to better suit the needs of the internetwork user. To this end, this thesis will develop the functional requirements for an "ideal" internetwork, describe two current methods for internetworking, and analyze these methods against the ideal internetwork. The advantages and disadvantages of each internetworking method will be discussed. After this analysis, suggestions will be made as to how these internetworking schemes could more closely resemble the "ideal" internetwork.

Contents

Chapter 1 - Introduction

Currently the world is full of computer networks, mainly working independently of each other. This is due to several factors. Often there is no company-wide strategy in place for the purchase of similar computer systems which can be networked together. This results in "islands" of information within a company, with little means for the exchange of information. Some computer systems run proprietary network software and this can present connectivity and interoperability problems. And lastly, physical location of a given computer network can lead to isolation of it from other computer networks and systems [12,20]. As owners of systems become more and more interested in sharing the resources of other networks, the need for interfaces between networks becomes more urgent. Internetworking was often an afterthought, if it was thought of at all. As a result, because of their inherent differences, many networks do not easily interface without additional functionality. This functionality can be present in the form of a specialized gateway computer. The gateway would be responsible for interfacing to potentially different physical media as as well as converting the protocols used by the differing systems.

Examples of the various differences between networks include:

- the physical medium used to carry the information.
- the format of the data carried on the physical connection.
- the method (protocol) for exchanging the data.
- the naming conventions used to identify other computers and networks.
- the quality of services offered by the network.
- routing methods

To cope with these differences, various schemes have been devised. "Bridges" have been designed to span networks which have differing physical link layers. "Gateways" have

been designed to span networks which have differing data link layers. "Protocol converters" have designed to span networks which have differing protocols in the upper 4 layers of the OSI reference model. None of these solutions are perfect. Their tradeoffs, advantages, and disadvantages will be discussed in the course of this thesis.

There are two major groups addressing the issue of internetworking. One of these groups is the Defense Advanced Research Project Agency (DARPA). It has developed what is currently known as the DARPA Internet. This internet has been in existence for over ten years. It has changed and evolved over the years, as a result of actual use and experiences. The other group involved in internetworking efforts is the International Standardization Organization (ISO). This is only part of their responsibility, which includes the specification of architecture and protocols which span all seven layers of the OSI network model.

The DARPA Internet has made a large impact in the area of internetworking. The Internet consists of approximately 2400 hosts, 400 networks, and many times that number of users [47,58]. "Requirements for Internet Gateways", RFC 985 [58], estimates the Internet growth rate at 10% per month. The protocols used by the Internet are commonly known as TCP/IP. The IP (Internet Protocol) is the network level protocol responsible for interfacing to the data link layer, and controlling the routing of data packets. The TCP (Transmission Control Protocol) is responsible for interfacing to the IP and offers the services of reliable data transfer and flow control. The Internet protocols are attractive for many reasons. One is that they have demonstrated their ability to perform reliably and efficiently, and are considered to be mature and stable [61].

The OSI protocols are not without merit. They may hold the greatest eventual potential for linking various networks into the world's largest internet. However, the OSI design committees have been faulted for not giving the experiences learned from the DARPA

Internet the weight they deserve [68].

Chapter 2 - Problem Description

The proposed thesis will have the following goals:

- To define an "ideal" internetwork, in order to analyze internetworking protocols and services.
- To describe how internetworking is handled by the DARPA Internet and OSI protocols.
- To compare the DARPA and OSI internetworking to the "ideal" internet.
- To propose a new set of internetworking protocols, or enhance existing protocols and services.

Throughout this thesis, the Open Systems Interconnect (OSI) Reference Model will be referenced. This is the well-known seven layer model, which specifies a different level of functionality for each layer of the model. This model can be applied to computer systems, and its lower three layers are concerned with various functions which contribute to internetworking. Where appropriate, a given protocol analysis may be mapped onto the seven layer model. This is done in order that the reader may gain an understanding of how one protocol compares with another.

In order to fulfill the first goal of this thesis, an internetwork will be defined. An internetwork, or internet, can be described as a network of networks. It is a result of the interconnection of two or more networks, often for the purpose of greater resource sharing, communications, and/or to provide a single administration point. To separate the notion of a single network from that of an internetwork, single networks will be referred to as subnetworks. It is therefore the case that internetworks are the result of interconnecting one or more subnetworks.

An "ideal" internet will now be defined. This definition will be a functional specification of what a user would ideally want an internetwork to offer. It should be realized that such an internetwork could probably never be built, as tradeoffs exist. This ideal internet will

represent a standard by which other internetworks can be analyzed. Since different users desire different services, fitness as a "good" internet will be dependent to some extent on the usage.

The qualities of an "ideal" internetworking design include the ability to:

- be easily extended.
- interface to as many different subnetworks as possible.
- function in the face of node/subnetwork/internetwork failure.
- maintain a level of performance acceptable to subnetwork users.
- provide various levels of data security.
- maintain data integrity.
- hide the specifics of the subnetwork interface from the user.
- allow for decentralized administration.
- handle the accounting of traffic (data).

The qualities of an ideal internet were derived from several sources [5,9,13,41,43,65,68,70,72]. After the ideal internetwork is defined, two well-known internetworking solutions will be examined. The internetworking schemes examined will be evaluated on the basis of the above criteria. From the analysis, weaknesses and strengths will be identified for each internetwork. Conclusions and recommendations will be made on the basis of these analyses.

The DARPA and OSI internetworking protocols will be examined as a second goal of this thesis. Internetworking has been extensively researched in the past 20 years. The implementation of ARPANET in the late sixties is a good example of this. This is a real life, working example of an internetworking solution. Over the last ten years, the DARPA Internet has grown extremely large. It consists of hundreds of machines, groups of which belong to various kinds of networks. For these reasons and because of the significance of it as a viable internetworking solution, it will be studied and analyzed

according to the ideal internetwork. The Internet protocols have evolved over the years, and the history of them and their design philosophy will be discussed.

The other internetworking scheme to be examined is that offered by the OSI protocols. The OSI architecture, services, and protocols have specified for each layer of the seven layer OSI Reference model. Although some of these protocols are still being discussed by committees, there is enough information available for an analysis. The protocols and architecture specified for the transport and network layers are those which have the most impact on internetworking design. The solution offered by the OSI protocols has perhaps the greatest potential for a following larger than any other internetworking design. Major computer vendors, third-party software houses, the U.S. government, and others have already begun to line up in favor of it. The U.S. government has stated that all software purchased in the near future must conform to the OSI protocols [69]. However, not everyone is a proponent of the OSI protocols. There are critics that claim the OSI committees did not give enough weight to the experiences learned in the DARPA community. Due to the potentially large impact of the OSI internetworking protocols, and because the effort is not without critics, it too will be analyzed and compared to the ideal internetwork.

Finally, this thesis will propose its own internetworking design. It will be the goal of this design to fit the specifications of the ideal internetwork in a way that is superior than either DARPA or OSI solutions. It is not expected that this design will be radically different from either the DARPA or OSI protocols, since one of the goals of the ideal internet is the ability to interface to existing subnetworks and internets. In fact, it may be that the proposed internet design is an enhanced version of one of the aforementioned internetworking schemes.

The Ideal Internet

This section will discuss the qualities and characteristics of an "ideal" internet. Each quality of the ideal internet will be outlined from a user and/or internetwork administrator perspective. The following are qualities that an ideal internet would possess:

- Extensibility
- Flexibility
- Interoperability
- Robustness
- High Performance
- Data Integrity
- Transparency
- Ease of Administration
- Traffic Accountability

Extensibility

Extensibility can be described as the ability to easily extend the internet. The addition of new subnetworks, gateways, and nodes will be hampered if the internet requires elaborate steps to be followed. It is expected that any new subnetwork added on will have to register with one or more central control points. This is at the very least necessary for reasons of routing and addressing. Given human nature, if it is difficult to add new members to the internet, few additions will be made, and this will limit the ultimate growth and usability of it.

Flexibility

This quality of an internet can be described as the ability of the internet to change as technology changes. For example, if a new physical medium technology becomes available and it is desirable from a user's perspective, then the internet should be able to accommodate it. Another example of flexibility is the ability to let users choose how to

configure their subnetwork. The users can decide where any given functionality should reside, based on cost and performance. Following the OSI reference model is one method for partitioning functionality, regardless of the vehicle used for executing it. To do less than this would almost ensure obsolescence, or at the very least limit the growth potential of the internet. It is more important to design for easy modification of the internetwork, than to put together a single optimized network design [9]. It is also important that the internet be able to employ new protocols as they become available. This is especially necessary if the bandwidth of the physical medium is increased, beyond the limits of the higher level protocols. Another aspect of flexibility is in the type of services offered by the internet. For example, a user may want to emphasize the quick delivery of data versus the accuracy of the data. Such is the case with digitized speech. In other cases the user may wish to emphasize the accuracy of the data, as opposed to the speed of delivery. This is usually the case with file transfers. The point here is that it is desirable to allow the user the ability to choose the service required, not to mandate it.

Interoperability

This quality of an internet can be described as the ability to interface to a variety of heterogeneous hosts and subnetworks. This quality becomes increasingly important as the needs of the user require the sharing of expensive resources, which reside on remote hosts and subnetworks. The difficulty with implementing this feature of an internet is that often a subnetwork vendor makes no attempts in the initial design of a subnetwork to include the ability to interface with foreign subnetworks. Currently, many vendors are moving toward the support of de facto standards such as TCP/IP. It can be envisioned that if all vendors already had compatible protocols in place, there would be little need to discuss this issue.

Robustness

This quality of an internet can be described as the ability to function in the event of node,

gateway, subnetwork, or internetwork failure. Node failure is inevitable, and at best can only be minimized. Even if a given piece of hardware is only inactive for preventative maintenance, these gaps in availability need to be addressed. From the user's perspective, it should be unnecessary to detect and re-establish communication with a remote node if an intermediate gateway or subnetwork fails or otherwise becomes unavailable. The internetworking scheme should handle this. There are various methods for providing this service, and the implementation can affect performance, and has a direct effect on the amount of state information that needs to be kept at the intermediate nodes.

High Performance

The performance of an ideal internet will not drop below the level users on the subnetwork expect. The actual performance realized will be dependent on the internet hardware and protocols, as well as the performance level of any intermediate nodes, gateways, and subnetworks. The ideal internet will allow delivery of data at a rate sufficient for even the most data-heavy applications. An example of a data-heavy application would be one involving the real-time delivery of digitized speech. Real-time digitized speech is interesting since late data is worse than incorrect data. These requirements for data delivery rate are much more stringent than those for file transfer. File transfer necessitates an accurate delivery system, as opposed to a quick one.

Data Integrity

To some internetwork service users, data integrity is an important quality. For those that advertise data integrity as a feature, they are obligated to insure the integrity of transmitted data. It should be noted that some subnetworking protocols do not insure data integrity (or delivery), and leave data integrity up to the higher layers (i.e. the transport layer). Data integrity can be accomplished on a hop by hop basis, or on an end to end basis, depending on the mode used by the network layer protocol.

Transparency

Transparency is the ability of the internet to be extended in a way that is uneventful to the rest of the internet. For example, it would not be appropriate for the entire internet to shut down for the addition of a new node, or in order to install new software on a node. Part of transparency is related to the robustness of the internet, i.e. how does the internet handle a situation when a node is coming up, going down, or simply not handling internet traffic at a given moment in time. In summary, the user on the internet should not be disturbed, or have his activities affected simply due to some modifications going on elsewhere in the internet. Another way to look at transparency is from the perspective of the actual data as it travels through the internet. It is quite possible that a given data packet will be fragmented, or broken up into smaller data packets as it travels from subnetwork to subnetwork. This is due to the fact that all networks do not have the same maximum size for a data packet. This necessitates that the packet be at some point broken up, then reassembled prior to delivery to the destination. The user at the source (or destination) node should not be concerned with this fragmentation or reassembly process. The ideal internet will exhibit a high degree of transparency.

Administrability

This quality of an internet addresses the ability of internet to be administered. An internet can be administered in two ways. One method of administration is for one central group (agency, university, research lab, company, or country) to administer the entire internet. This is the centralized approach. This has the advantage of a single focal point controlling the administrative functions. There would be no arguments or agreements necessary; whatever the central administration did, would go. This has the distinct disadvantage that there are probably no groups that would care to have control of their subnetwork relinquished to an external group, especially one in another country. The other method of internet administration is a decentralized one. This method involves each subnetwork site doing its own administrative functions. This is attractive as a site

can administer itself any way it wants to. Of course, there are certain functions which will require coordination between sites if a given subnetwork is to belong to the internet.

There are several facilities which an ideal internet administration package would allow for. First, it would allow for debugging of faults which are detected. This would include functionality to report on faults as they occur, a method for logging these reports, and a method for the execution of diagnostic tests [39]. Another part of administration of an internet is the ability to configure a given subnet, or domain. This includes naming of nodes in the local subnet, reconfiguring the topology, and communicating these changes back to the internet.

Traffic Accountability

This quality of an internet is the ability to charge a given user for the passage of data through a subnet and/or for the use of services. This information would have sufficient granularity so that it could be determined how much to bill a given user. This would be based on amount of data, time of day it was handled, type of service rendered, and the length of time necessary to transfer the data.

Chapter 3 - Thesis Limitations

This thesis will not attempt to prove any of the protocols it suggests. That is, there will be no implementation to test, in actual use, the enhancements suggested. However, for any enhancements, there will be a desciption of how it would operate, along with any advantages and disadvantages. An implementation of a suggested protocol could be accomplished as a separate thesis effort.

There will be no attempt made to examine all the schemes which currently exist to do internetworking, although it is recognized that others do exist. The DARPA and OSI protocols have been chosen because of their current widespread use and potential future use, respectively. Examples of other internetworks include JANET, ROSE, the Xerox Internet, the AT&T Internet, DEC Easynet, CAMPUSNET [11], NRI (National Research Internet), and BITNET.

The NRI is a collection of Federal agency-owned subnetworks. The AT&T Internet is administered and used internally by AT&T; it is TCP/IP based. DEC Easynet is based on DECnet, and is used internally by Digital Equipment Corporation. JANET is the Joint Academic NETwork, based in the United Kingdom. ROSE is the Research Open Systems for Europe; it is being used as a proving ground for the ISO protocols. BITNET is the Because It's Time NETwork; it consists mainly of University members. A brief description of these internets is contained in [47].

Chapter 4 - The OSI Approach to Internetworking

Before attempting to describe the methods for internetworking developed by ISO, a brief history leading to their evolution will be given. The ISO, seeing a need for a unified approach for computer communications, developed what is commonly known as the seven layer OSI reference model. This model partitions the task of interconnecting computer systems into seven distinct and well-defined layers. The ISO standards documents specify this model fully [25]. From highest to lowest layer, the layers defined by the reference model are:

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

Application Layer

This layer supplies those services commonly needed by the user applications layer services. This typically can include such things as virtual terminal, file transfer, and electronic mail services.

Presentation Layer

This layer supplies services which are required by sufficiently many applications, that a general solution is indicated. Examples of this include data representation and compression. Take for instance, the problem of exchanging integer data on end machines which encode integers in different ways (i.e. 68000 vs. VAX). It is the job of this layer to handle these differences in a way which is transparent to the application layer.

Session Layer

This layer allows end systems to establish sessions with each other. This is useful for synchronizing a dialogue between the two systems. For example, where there is a half-duplex connection between end systems, the session layer services could keep track of whose turn it is to transfer data on link. Synchronization services may also include remembering where a transaction left off when a crash occurred. This is useful when doing a lengthy file transfer. This feature, commonly known as "checkpointing", would allow the transfer to resume at approximately the point of the last crash.

Transport Layer

The services offered by this layer include the transportation of data from end system to end system. If need be, the data can be segmented by this layer and later reassembled at the destination end system. The transport layer can also do multiplexing, in order to speed up delivery and increase throughput. This all occurs transparently to the session layer. This is also the lowest layer in the model that is considered to be involved in end-to-end communications with its peer entity. That is, the layers beneath it only communicate with the next intermediate system, which may not necessarily be the destination end system.

Network Layer

This layer provides high level access and control of the subnetwork upon which it operates. This is the layer responsible for the routing of data packets. This layer may also provide congestion control, accounting, and communication facilities over homo/heterogeneous subnetworks. As will be shown later, this layer has its own mini-architecture, consisting of three sublayers. This is due in part to the diversity and complexity of the various subnetworks and the methods which can be used to interconnect them.

Data Link Layer

This layer provides an error free communication service across a given subnetwork. To accomplish this, it takes a primitive physical transmission facility (offered by the physical layer) and transforms it into a reliable transfer medium. This occurs transparently to the network layer.

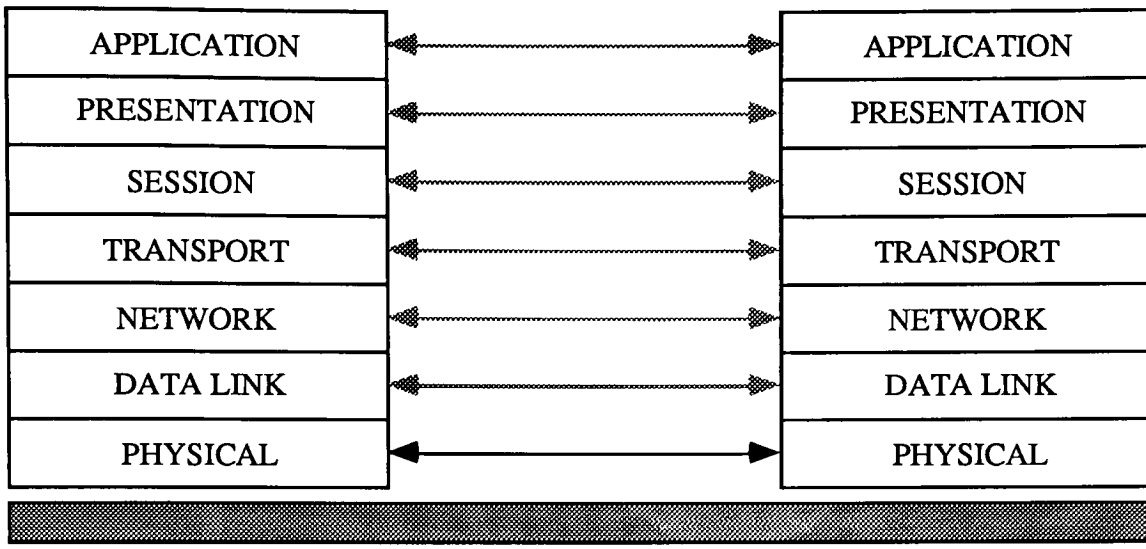
Physical Layer

The physical layer provides the service for sending raw bits over the physical medium. This layer deals with the issues of procedural, electrical, and mechanical interfaces to the physical transmission medium [68].

The OSI reference model, like all ISO standards, was discussed and agreed upon by international committee. The seven layer partitioning came about by following these general design guidelines [40]:

- the overall architecture should be simple
- keep interfaces between layers simple
- functions which are very different belong in separate layers
- combine into the same layer those functions which are similar in nature
- use past experience to select boundary points between layers
- maximize ability for change by creating layers where internal mechanization can be made independently from the functionality
- create layer boundaries where the requirement may exist in the future
- create layers where different levels of data abstraction are required
- create layer boundaries with only the layer above and the layer below
- allow sublayering to occur where communication services require it
- allow the by-passing of sublayers

The following diagram illustrates the position of each layer with respects to the others:



The dotted lines indicate a logical connection between the layers, while a solid line indicates a physical connection between the layers. Each seven layer "stack" is meant to show the architecture an end or intermediate system would implement.

Sect. 4.1 - The OSI Standards for Internetworking

This section will summarize the OSI standards specified by international committee thus far that involve internetworking. The section following this will detail the services offered by the network and transport layers. Examples will be provided of how a service is derived from an actual protocol. The options open to an OSI implementor are more diverse than those offered by the DARPA protocols. The main difference is that the OSI protocols allow both connection-mode and connectionless-mode network level services. With TCP/IP internetworking, the network service is strictly connectionless. Thus the description of the TCP/IP network services is a more straightforward one.

In order to describe the services and provisions for these services at the network layer, several OSI standards have been issued:

- ISO 8648 describes the internal organization of the network layer. This document details the three layer mini-architecture that exists there.
- ISO 8348 describes the connection-mode network service definition.
- ISO 8348/AD1 (Addendum 1) describes the connectionless-mode network service definition.
- ISO 8348/AD2 describes the network layer addressing details.
- ISO 8880 Part 1 describes the general principles and conformance for providing the network layer service.
- ISO 8880 Part 2 describes the provisions for supporting connection-mode network service.
- ISO 8880 Part 3 describes the provisions for supporting the connectionless mode network service.

Sect. 4.2 - The Network Layer Architecture

The network layer requires its own mini architecture because of the wide variability of subnetwork services and technology. In order to allow for changing technology and ease of interfacing, the network layer has been divided into three sublayers [40,66,68]:

- SNICP - the SubNetwork Independent Convergence Protocol
- SNDCP - the SubNetwork Dependent Convergence Protocol
- SNAcP - the SubNetwork access Convergence Protocol

SNICP

This protocol is built on the most minimal set of services provided by a given subnetwork access protocol. The idea here is to build a single protocol which may be used universally by all the end systems and internetworking units (gateways) on the internet. It is therefore known in advance, that this protocol will not require services of any subnetwork, that the subnetwork will not be able to provide. ISO 8348/AD1 (ISO IP) is an example of an SNICP.

SNDCP

This protocol will either provide the OSI network service, or possibly be used to provide the service required by the SNICP. This protocol may be used to enhance a given subnetwork service and bring it up to the level of service required by the network layer. An example of this is the use of an SNDCP with the 1980 version X.25. This version of X.25 is deficient in the ability to transfer information regarding ISO network connection establishment [66]. ISO has defined a standard, ISO 8878, which describes how one would implement the SNDCP required to bring this version of X.25 up to the level required by the OSI network layer services.

SNAcP

The protocol at this level is required to interface directly to the subnetwork in question. It

is not required to provide the network layer service on its own. It is usual that the SNICP, or a combination of SNICP/SNDCP, coupled with the SNAcP come together to fulfill the requirements of the network layer.

As is the case with the OSI reference model, the mini architecture for the network layer is not in itself a specification of any given protocols. As will be shown in the next section, it is sometimes the case that a single protocol (1984 X.25) can be used to realize all three sublayers and provide the OSI network layer service.

Sect. 4.3 - General Network Layer Services

The network layer of the ISO reference model is very complex. This is primarily due to the diverse subnetworks and protocols which reside at this level. In general, the network layer protocols provide a means for transmitting Network Service Data Units (NSDUs) across a subnetwork, or a network of subnetworks. This strongly implies the need for a routing function as well. When the first standards began to emerge for this layer, they were connection-oriented. Later, standards providing connectionless services were developed.

The OSI standards for connection-mode operation describe the network layer as providing the means of establishing a connection, transferring NSDUs between transport entities, and terminating connections. The connectionless-mode operation need only provide the means for data transfer. The network layer provides the transport layer with a transparent view of data transmission across potentially heterogeneous networks. Routing is therefore an important function of the network layer.

General Characteristics of the Network Services

The Network Service (NS) provides for transparent transfer of data between NS users. Specifically, the NS provides [32]:

- Independence from the underlying subnetwork technology. This shields the user from the peculiarities of a given subnetwork interface or interfaces.
- End-to-end transfer. The network service (connection-mode) provides for end-to-end data transfer between NS users.
- Transparency of transferred data. The NS makes no attempt to interpret or restrict the content of the data between NS users.
- Service and quality selection. The NS provides a means by which the NS user may agree upon the level of service quality for the transfer of data. Quality of Service (QOS) parameters include throughput, transit delay, accuracy, and

reliability.

- NS user addressing. The NS provides a means of uniquely identifying a given NS user, using a Network Service Access Point (NSAP) address.

The services and functions of the network layer include:

- addressing: The transport layer must supply a unique name, so as to identify the location of the process at the destination host.
- expedited data service: this denotes the ability to deliver data in a prioritized manner. This means expedited data may be delivered ahead of previously sent "normal" data. The intent here is to offer a service by which priority information can bypass flow control facilities at a given Network layer entity.
- reset service: This allows the transport entity to abort the current communication processing. This would allow data transmission from the point of reset, after synchronizing the end system transport entities.
- routing and relaying: The process of sending data from one system to another may necessitate a multiple trip through one or more different subnetworks.
- segmentation: This involves the breaking up of NSDUs which are too large to pass through a given subnetwork. The offering of this service implies the availability of a reassembly service as well. This service provides the means of putting the original NSDU together, from its segmented parts.
- flow control: This service helps to prevent data from being exchanged in a manner that overwhelms the end or intermediate systems.
- error control: This service gives the transport layer a means for determining problems in the network layer entities which may reside between the end systems.
- type of service (TOS): This service allows the transport layer to specify the type of service it requires. Since not all subnetworks support the wide range of quality

expected, the network layer may be enhanced to provide the required degree of quality.

Sect. 4.4 - OSI Internetworking Strategies

There are basically two strategies for the internetworking of OSI systems [66]:

- Hop-by-Hop
- Internet Approach

Hop-by-Hop

This approach takes the individual subnetwork and enhances it to the level required by the network layer service. As an example, an SNDCP would be used to implement the features needed by the network layer, and make up for anything lacking in the subnetwork services. The 1984 version of X.25 is an example of a subnetwork access protocol which implements the network layer services without the need for a SNDCP. The 1980 version of X.25 is an example of a subnetwork access protocol which requires the use of an SNDCP to bring it up to the level required by the network layer services.

Internet Approach

This approach involves the building a single SNICP, which would include all the enhancements needed by the subnetwork with the most minimal of services.

The hop-by-hop approach has the advantages of taking advantage of the services offered by a given subnetwork. On the other hand, the disadvantage of this is that a different SNDCP is required for every different SNAcP. The advantage of the internet approach is that one protocol can be used with all SNAcPs. The disadvantage of this approach is that it assumes only minimal functionality from the SNAcP, and does not take advantage of built in services as they exist in the various subnetworks.

OSI Network Connection Types

The network connection services offered by the network layer services can be grouped into two categories: connection-oriented and connectionless. The remainder of this section will be devoted to describing the general characteristics of both connection types.

The next section will describe the services offered by both flavors of the OSI Network Service.

Sect. 4.5 - Connection-oriented Network Service (CONS)

This category of network connection contains the following provisions:

- a connection which may be used by the network service users for the transfer of data.
- a level of quality across the connection that may be negotiated by the network service providers and users.
- the exchange of NSDUs over a connection.
- expedited data, or the ability to specify special control over certain NSDUs.
- a reset service which provides the ability to abort and resume a connection in a controlled manner.
- a data confirmation service.
- a network connection termination service.

The CONS approach involves the creation of a logical path, called a virtual circuit, over which the NSDUs will be transmitted. Each intermediate system will have a logical connection to the next system in the path to the end destination system. Once the virtual circuit is established, all the NSDUs transferred between the end system will traverse the same path. It should be noted that since the CONS is merely an interface, it is possible that the underlying subnetwork service is inherently connectionless. It is not, however, clear what the advantages would be of arbitrarily providing the CONS in this matter. This method has several advantages. First, once a virtual circuit is established, it is no longer necessary to consider the end system address in the NSDUs. An intermediate system need only look in its tables as to determine where data coming in on virtual circuit X needs to be sent out on. This simplifies the routing functionality needed, as well as the amount of processing resources needed to route the NSDUs. Also, since a single path is used to link the end systems and the intermediate systems, there is no resequencing required. The data is received at the end system in the order it was sent.

Flow is under more control using CONS. This allows the intermediate systems to throttle data accurately so that an intermediate or end system is not overwhelmed.

Connectionless Network Characteristics

The CLNS had its beginnings in datagram style networks, such as ARPANET [40]. As has been shown in previous chapters, these types of networks rely on the user (i.e. transport layer or above) to ensure data integrity. The ISO has issued a standard for the specification of a connectionless-mode network service [32].

Connection-oriented Network Service (CONS)

The standard for CONS is specified in ISO 8348 [32] and is described in [40,45,66,68]. This standard defines the network service with regard to:

- the primitive actions and events of the service
- the parameters associated with each primitive action and event.
- the interrelationships between valid sequences of the events

A major goal of this standard is to provide a set of services that can be applied to existing heterogeneous subnets. This could potentially allow the interconnection necessary for global communication.

Background

The network service provides a means for the transparent transfer of data (NS-user-data) between NS users. It is meant to provide this service in a way that does not reveal the techniques used by the underlying subnetwork services. This is an important advantage, as these subnetwork services do not generally possess a common interface, between vendors.

There exists a set of parameters which can be used to define the quality of service between two NS users. For a given phase of a connection, there are parameters which

specify either speed or accuracy/reliability [66]:

Phase	Speed	Accuracy/Reliability
connection establishment	NC establishment delay	NC establishment failure probability
data transfer	throughput transit delay	resilience of transfer NC transfer failure probability
connection termination	NC release delay	NC release failure probability

NC Establishment Delay

This value represents the maximum allowable delay between an N-Connect.request and the corresponding N-Connection.confirm.

NC Establishment Failure Probability

This represents the ratio of connection establishment attempts that fails as a result of NS provider malfunction. This includes misconnection, NS refusal, and excessive delay.

Throughput

This represents the amount of NSDU transfer that the provider can ably sustain. Desired and maximum allowable values can be specified.

Transit Delay

This value represents the average elapsed time between an N-data.request and the corresponding N-data.indication. Desired and maximum acceptable values are specified.

Residual Error Rate

This equates to: (incorrect NSDUs + Lost NSDUs + duplicate NSDUs) divided by Total NSDUs transferred.

Transfer Failure Probability

This value pertains to throughput, transit delay, and residual error rate. This represents the observed proportion of time that the NS provider fails to provide the minimum acceptable service.

NC Resilience

These parameters specify the probability of an NS provider invoked NC release, and NS provider invoked reset.

NC Release Delay

This represents the maximum acceptable delay between an NS user invoked N-Disconnect.request and the successful release of the NC at the peer NS user.

NC Release Failure Probability

This is the ratio of total NC release requests that result in failure to the total number of release requests issued.

NC Protection

This parameter is used to specify the extent to which an NS provider attempts to prevent unauthorized tampering, misauthentication, or monitoring of NSDUs. There are 4 features which may be selected:

- confidentiality
- modification, deletion, replay, or insertion of NSDUs
- NSDU origin authentication

NC Priority

This parameter specifies the relative importance of gaining/keeping an NC connection, and the priority of data on the NC.

Maximum Acceptable Cost

This represents the maximum acceptable cost for an NC. This may be specified in absolute or relative cost units.

Connection-mode Primitives

The following list represents the service primitives and associated parameters, for a given phase of connection:

NC Establishment

- N-Connect.request (called address, calling address, receipt confirmation selection, expedited data selection, QOS parameter set, NS-user-data)
- N-Connect.indication (called address, calling address, receipt confirmation selection, expedited data selection, QOS parameter set, NS-user-data)
- N-Connect.response (responding address, receipt confirmation selection, expedited data selection, QOS parameter set, NS-user-data)
- N-Connect.confirm (responding address, receipt confirmation selection, expedited data selection, QOS parameter set, NS-user-data)

Data Transfer

- N-Data.request (NS-user-data, confirmation request)
- N-Data.indication (NS-user-data, confirmation request)
- N-Data-acknowledge.request ()
- N-Data-acknowledge.indication ()
- N-Expedited-data.request (NS-user-data)
- N-Expedited-data.indication (NS-user-data)
- N-Reset.request (reason)
- N-Reset.indication (originator, reason)
- N-Reset.response ()

- N-Reset.confirm ()

NC Release

- N-Disconnect.request (reason, NS-user-data, responding address)
- N-Disconnect.indication (originator, reason, NS-user-data, responding address)

The proceeding sections will describe each of the three phases of a connection. Included will be a description of the service primitives used and the parameters associated with the primitives.

Network Connection Establishment Phase

The NC establishment service primitives are used to establish an NC between NS users. The NS users must exist and be known to the NS provider. Simultaneous N-Connect requests are handled separately and may result in zero, one, or two NCs.

The following section details each parameter associated with connection establishment.

Background on Addresses

There are three network connection parameters which are considered as addresses:

- called address
- calling address
- responding address

These are all regarded as network service access points (NSAPs). The NSAP address parameters accommodate variable length addresses, up to a maximum of 40 decimal digits. The values contained in these parameters are not necessarily checked or authenticated by the NS provider.

Called Address

This parameter conveys an address identifying the NSAP to which an NC is desired.

Calling Address

This parameter conveys the address of the NSAP from which the NC has been requested.

Responding Address

This parameter conveys the address of the NSAP to which the NC has been established.

Receipt Confirmation Selection

This parameter selects whether or not to use or make available the receipt confirmation service. This service is not mandatory and therefore need not be provided by the local, or intermediate, NS provider(s).

Expedited Data Selection

This parameter is used to select the use/availability of the expedited data transfer service.

QOS Parameter Set

All QOS parameter sets exchanged during NC establishment has associated with it the following sub-parameters:

- target value (value desired by the calling NS user)
- lowest quality acceptable (lowest value acceptable by the NS user)
- available value (value NS provider is willing to provide)
- selected value (value the called NS user agrees to)

Throughput and transit delay are currently the only QOS sub-parameters that are negotiated during NC establishment. All other QOS sub-parameters are currently provided for in a manner not specified by the network service standard.

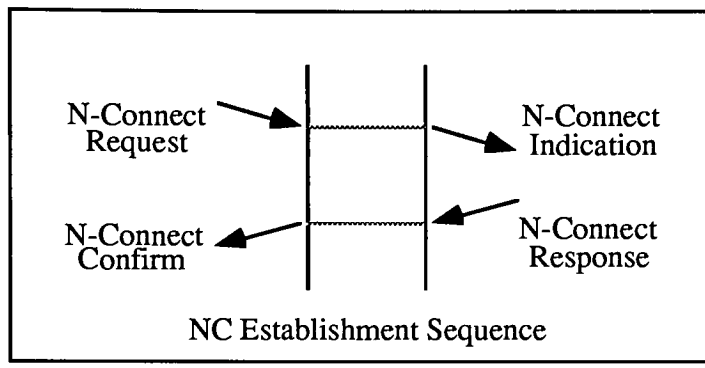
NS-user-data

This parameter actually stores the data being exchanged by NS users. The amount of data per NSDU can be up to (and including) 128 octets. It should be noted that this is not the

normal parameter for transfer of data. Its use is mainly for the transfer of set up data.

Sequence of NC Primitives

The following diagram shows the sequence of events for successful NC establishment:



NC Establishment Sequence

Note: for an explanation of the event sequence diagrams, please refer to Appendix A.

This connection establishment may fail due to the NS provider being unable to set up the connection. It may also fail due to the refusal of the called NS user to accept the N-Connect.indication. Also, the NC establishment may be aborted by the NS provider, or either of the two NS users prior to the issuing of the N-Connect.indication.

Network Connection Release Phase

The NC release primitives are used to release an NC. The release can be performed by either of the two NS users. The NC may also be released by the NS provider. The called NS user may issue an N-Disconnect as a way of rejecting an N-Connect.indication. The NS provider may also issue a release if it cannot provide the desired NC. Once an NC release is invoked at one endpoint, all remaining data and requests for the other endpoint may be discarded by the NS provider.

NC Release Parameters

The parameters associated with the NC release primitives will be described below.

Originator

This parameter indicates the source of the NC release. It indicates either an NS user, NS provider, or has the value "undefined".

Reason

This parameter conveys information with regard to the cause of the NC release. The following values are allowed for this parameter:

Originator = NS provider

- disconnection - permanent condition
- disconnection - transient condition
- connect rejection - NSAP address unknown/permanent condition
- connect rejection - NSAP unreachable/transient condition
- connect rejection - NSAP unreachable/permanent condition
- connect rejection - QOS unavailable/permanent condition
- connect rejection - QOS unavailable/transient condition
- connect rejection - reason unspecified/permanent condition
- connect rejection -reason unspecified/transient condition

Originator = NS user

- disconnection - normal condition
- disconnection - abnormal condition
- connection rejection - permanent condition
- connection rejection - transient condition
- connection rejection - QOS unavailable/transient condition
- connection rejection - QOS unavailable/permanent condition
- connection rejection - incomplete info in NS user-data

Originator = undefined

- reason parameter always equals undefined in this case

NS-user-data

This parameter allows for the transfer of data between NS users. An NS user which calls the release function is able to send 0 to 128 octets of data along with the NC release.

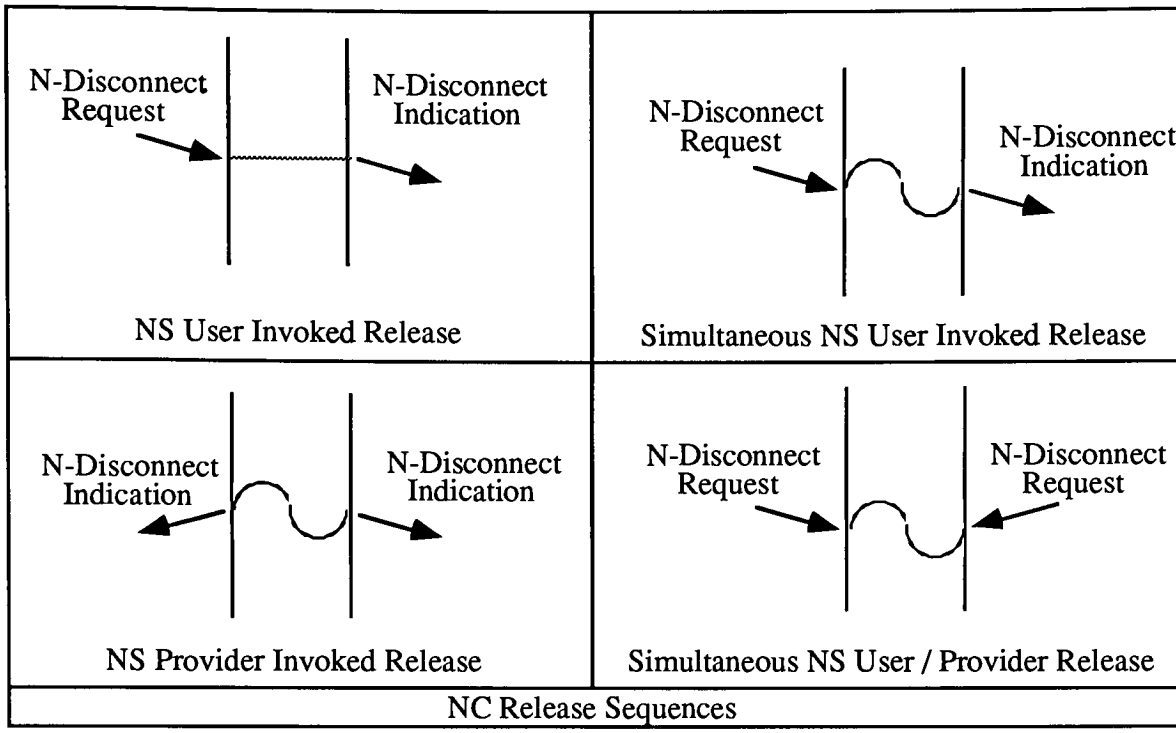
Responding Address

This parameter is present in the case where rejection of an NC occurs by the call from the NS user. This parameter contains the NSAP address from which the rejection was issued.

Sequence of Events for Releasing an NC

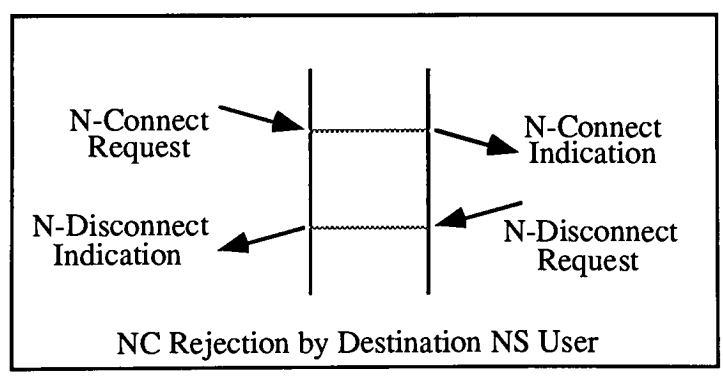
The following are valid sequences for the release of an NC:

- one NS user requesting release and the other NS user receiving an NC release indication
- both NS users simultaneously invoke NC release
- the NS provider invokes NC release, with NC release indications going to both NS users
- both NS user and NS provider release the NC by requesting NC release and indicating NC release, respectively.



NC Release Sequences

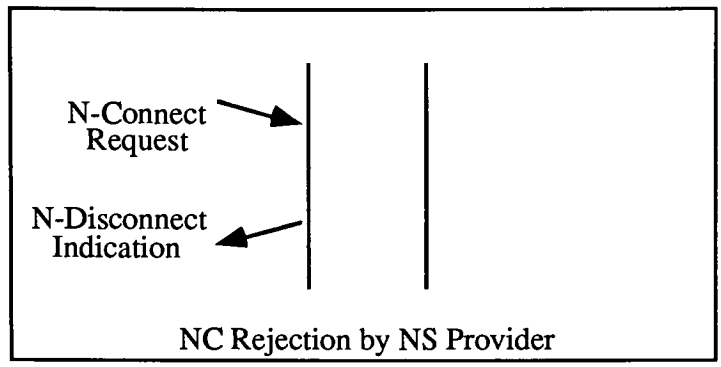
An NC connection may be rejected by the other NS user. In this case, NS user A issues an N-Connect.request and the other NS user, B, receives an N-Connect.indication. NS user B does not want the connection, and issues an N-Disconnect.request, and NS user A receives the corresponding N-Disconnect.indication.



NC Rejection by Destination NS User

An NC may also be released by the NS provider. This case can be illustrated by an NS user issuing an N-Connect.request and receiving back an N-Disconnect.indication. The originator parameter in the disconnect indication would show that the NS provider

rejected the connection:



Data Transfer Phase

The data transfer phase provides for the transmission of data between NS users. The data is transferred in the form of NSDUs, and may be sent in either direction, simultaneously, on the NC.

Data Transfer Primitives

The following primitives exist for providing the data transfer function:

- N-Data.request
- N-Data.Indication

Data Transfer Parameters

The following lists out each parameter associated with the data transfer primitives, and a description of the parameter's function.

NS user-data

This is the parameter that allows the data to be transferred between NS users.

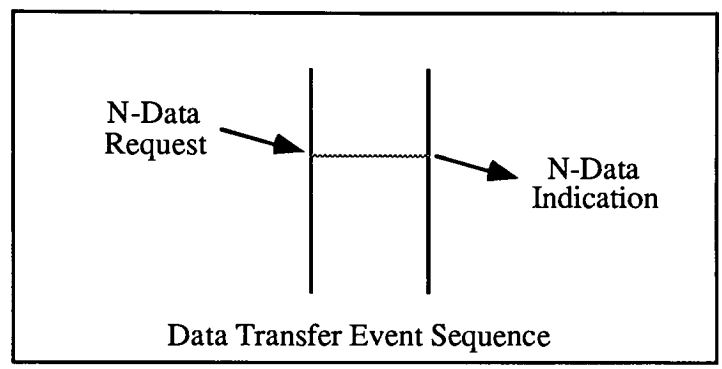
Confirmation Request

This parameter, when set, conveys to the destination NS user that a confirmation receipt

is desired by the sender. The N-Data-acknowledge primitives provide a method for such confirmation. This parameter is used only for NCs where negotiation for confirmation service was agreed to during the NC establishment phase.

Sequence of Events for Transfer of Data

The sequence of primitives in a successful data transfer is shown in the following diagram:



Data Transfer Event Sequence

Receipt Confirmation Service

The receipt confirmation service is used in association with specification of the service in the N-Data.request. Each NSDU sent in an N-Data.request with receipt confirmation set is acknowledged with an N-Data-acknowledge.request. These confirmation of receipt (COR) will not be combined with previous or future CORs; they are sent back to the originating NS user in the same order as the NSDUs are received. In this way, the originating NS user may count acknowledgments to determine which NSDUs have been acknowledged. Again, the receipt confirmation service is negotiated at NC establishment phase by the two NS users and the NS provider. It is not mandatory that the NS provider provide this service.

The following primitives are supplied for use in the receipt confirmation service:

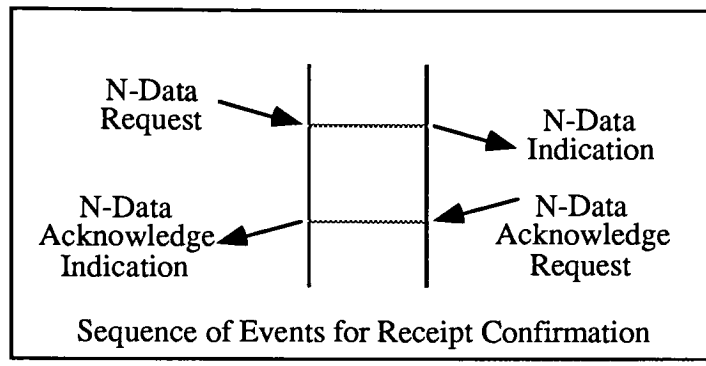
- N-Data-acknowledge.request

- N-Data-acknowledge.indication

There are no parameters associated with these primitives.

Sequence of Events for Receipt Confirmation

The following diagram illustrates the sequence of events for receipt confirmation of a successful NSDU transfer:



Sequence of Events for Receipt Confirmation

An NS user is not allowed to acknowledge an N-Data indication if the confirmation receipt is not set. Further, an NS user is not to send a second confirmation receipt on an N-data.indication which has already been acknowledged.

While in the process of acknowledging N-Data.indications, an NS user may receive an N-reset indication or confirm. At this point, the NS is not allowed to acknowledge any N-Data indications which it received up until the time the reset procedure completed.

Expedited Data Transfer Service

The expedited data transfer functions as an alternate method for sending data on an NC. The same sequencing rules for NSDUs apply to ENSDUs (Expedited NSDUs). The NS provider guarantees they will be delivered in the same order as they were sent. These ENSDUs have the special properties of being accepted by the receiving NS user ahead of NSDUs which may have previously arrived, but have not yet been processed. Also, the

ENSDUs can be accepted by an NS user which is no longer accepting NS-user-data. The use of this service is negotiated during the NC establishment phase. It is not mandatory for the NS provider to provide this service.

Expedited Data Transfer Primitives

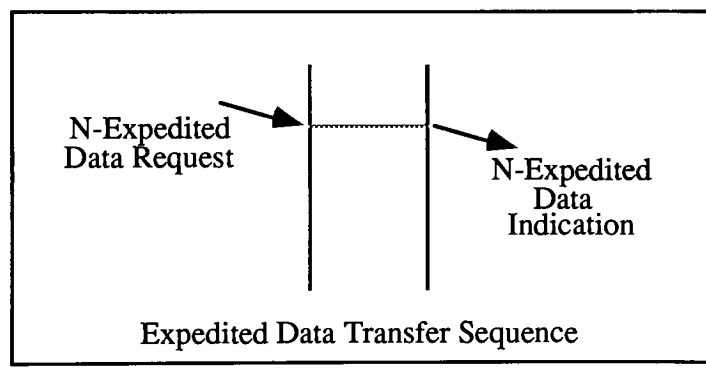
The following primitives are used to realize the expedited data transfer service:

- N-Expedited-data.request
- N-Expedite-data.indication

Both primitives have associated with then a single parameter: NS-user-data. The size of the NS-user-data may range from 1 to 32 octets.

Sequence of Events for Expedited Data Transfer

The following diagram illustrates the sequence of primitives for the expedited data transfer function:



Expedited Data Transfer Sequence

The Reset Service

The reset service may be used for two reasons:

- by an NS user to synchronize use of the NC
- by the NS provider to report loss of data which is unrecoverable within the network service

Reset Service Primitives

The reset service offers the following primitives to carry out the reset function:

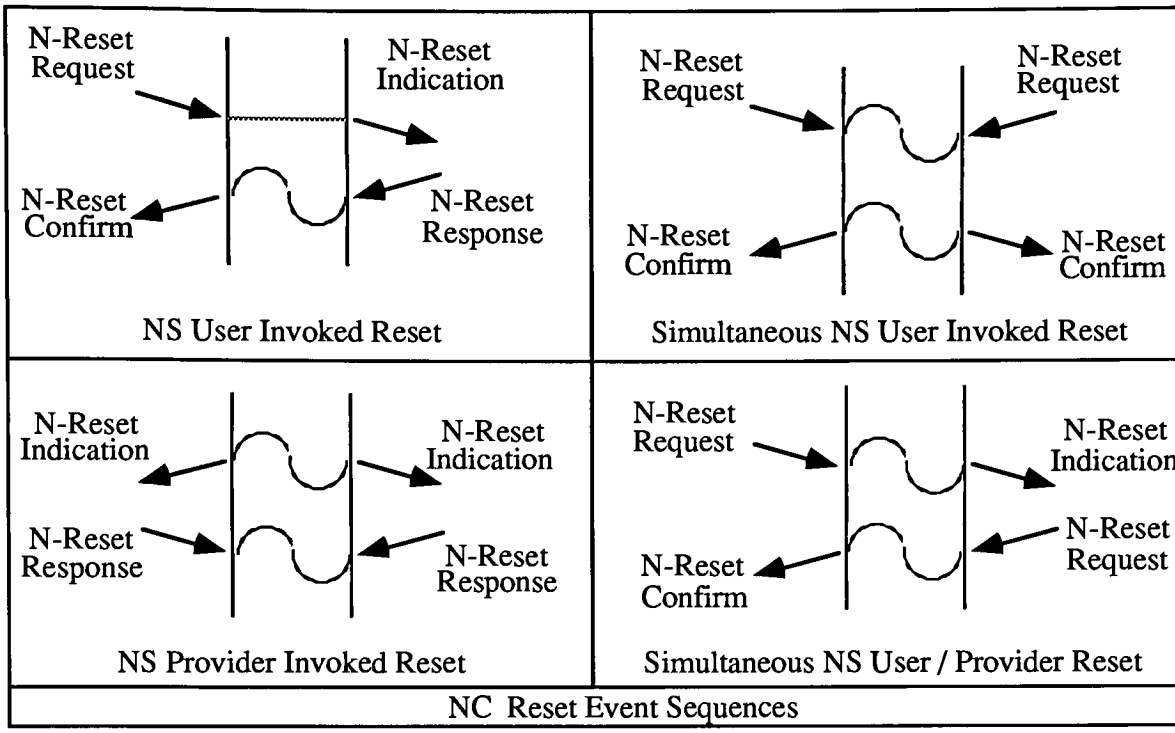
- N-Reset.request
- N-Reset.indication
- N-Reset.response
- N-Reset.confirm

There are two parameters associated with the reset service: originator and reason. The originator parameter is the value used to indicate the source of the reset. Possible values are "NS user", "NS provider", or "undefined". The reason parameter helps to give the reason for the reset. If the originator of the reset is the NS provider, then the reset reason can either be "congestion" or "reason unspecified". When the NS user invokes reset, the reason is simply "user synchronization". When the originator value is "undefined", the reason value is also "undefined".

Event Sequence for Reset

There are two sequences whereby NS user and NS provider exchange N-Reset primitives:

- An NS user issues an N-Reset.request and the NS provider responds with an N-Reset confirm
- An NS user receives an N-Reset.indication from the NS provider, then issues an N-Reset response to the NS provider.


N-Reset Request
N-Reset Indication
N-Reset Confirm
N-Reset Response
NS User Invoked Reset
N-Reset Request
N-Reset Request
N-Reset Confirm
N-Reset Confirm
Simultaneous NS User Invoked Reset
N-Reset Indication
N-Reset Indication
N-Reset Response
N-Reset Response
NS Provider Invoked Reset
N-Reset Request
N-Reset Indication
N-Reset Confirm
N-Reset Request
Simultaneous NS User / Provider Reset
NC Reset Event Sequences

Sect. 4.6 - Connectionless Mode Network Service (CLNS)

This section will detail the services provided by the network layer for the purpose of providing a connectionless mode network service. The standard for the CLNS is specified in ISO 8348 Addendum 1. It is interesting in itself that this service definition is an addendum and not part of the original network service specification. The standard mentions that it became necessary to define a connectionless mode network service. There are no specific reasons given for the inclusion of this service, except that they felt the connection-oriented service unnecessarily limited the scope of the reference model. The inclusion of the service in the network service definition will be discussed in greater detail in the OSI analysis section.

Background

The CLNS does not require a connection as does the CONS. Therefore, there is no clear relationship between units of data. Each unit of data may be routed separately, and may arrive at the final destination in an order different from that which was sent. Because there is no connection set-up, and the associated overhead for such activity, the CLNS is especially attractive for applications which require short-term request/reply transactions.

Quality of Network Service

There is no peer-to-peer negotiation of quality of service at the time the transmission of data takes place. This is by definition of the connectionless-mode service as described in ISO 7498 [25]. A knowledge of the quality of service characteristics are expected to be known prior to the transmission, between the NS users and the NS provider. Knowledge of the connectionless-mode characteristics are available to a sending NS user via a control or management facility. The following quality of service parameters are defined for the CLNS:

- Transit Delay
- Protection from unauthorized access

- Residual Error Probability
- Priority

Transit Delay

This parameter is the elapsed time between an N-Unitdata.request and the corresponding N-Unitdata.indication. This value is calculated only on NSDUs that are successfully transferred. Success is determined by the NSDU leaving the originating NS user and arriving at the destination NS user. Transit delay can change dynamically and is kept track of by the NS provider. The value is available to the NS user and can be evaluated prior to sending the NSDU.

Protection From Unauthorized Access

This QOS parameter allows the NS user to specify to the NS provider to choose the lowest cost method for transport. It may also specify the maximum allowable cost.

Residual Error Probability

This parameter denotes the probability that a single NSDU will be lost, duplicated, or delivered incorrectly.

Priority

This parameter allows the NS user to specify the relative priority of an NSDU. This priority may have an effect on which NSDUs are to have their quality of service degraded, or which ones are likely to be discarded if resources are in short supply.

Connectionless-mode Primitives

These functions serve to transfer NSDUs from NSAP to NSAP. All NSDUs are transferred independently of each other. There is no maintenance of state information between a pair of NSAPs, and the flow control available exists only between the NS user and NS provider interface.

There are two primitives associated with the CLNS:

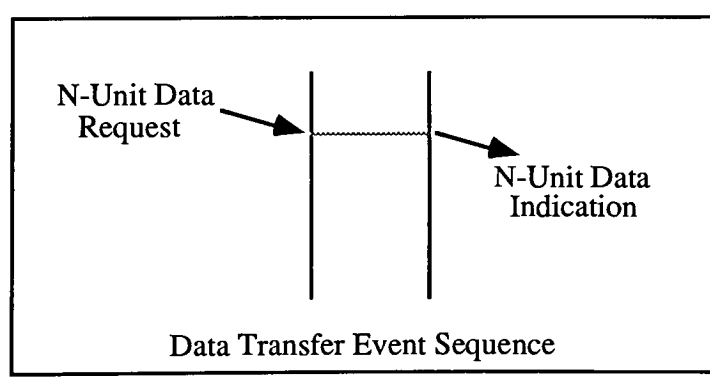
- N-Unitdata.request (source address, destination address, quality of service, NS-Userdata)
- N-Unitdata.indication (source address, destination address, quality of service, NS-User data)

In addition to the service primitives, the following parameters are associated with them:

- Source Address -This parameter specifies the NSAP of the originating NS user. This address is formatted identically in the OSI CONS.
- Destination Address This parameter specifies the NSAP of the destination NS user. This address is formatted identically in the OSI CONS.
- Quality of Service this value specifies the values for each of the QOS sub-parameters. For the request primitive, the values can be anything which is defined. For the indication primitive, these values may be different than those listed in the request primitive.
- NS-Userdata - This parameter allows the transfer data between NS users. Any number of octets, from 0 to 64,512, may be sent.

Event Sequence for Data Transfer

The following diagram shows the sequence of events for N Unitdata.request and N-Unitdata.indication:



Data Transfer Event Sequence

Sect. 4.7 - Protocols for Providing the OSI Network Service

The next 2 sections will describe the protocols for providing the OSI network layer services. The first section will describe the protocols which have been standardized by the ISO for providing the connection-mode network service. The second section will describe the protocol for providing the connectionless-mode network service. From a survey of the standards documents [33,36,37] which describe protocols for providing the network service, there is an attempt to describe the use of existing network and data link protocols. These are meant to serve as guidelines, and in no way mandate the use of a given protocol.

Sect. 4.8 - A Connection-mode Network Protocol (X.25)

There is an ISO standard, ISO 8878 [36], which describes the use of several protocols for providing the connection-mode network service. Most notably, X.25 in both the 1980 and 1984 versions are illustrated in these documents. This section will briefly describe the operations of the X.25 protocol, followed by the methods outlined by the standards documents for deriving the OSI network service from it. For a more detailed description of the X.25 protocol, the reader is encouraged to examine the ISO and CCITT standards [31,71].

The X.25 Protocol

The X.25 protocol is widely used as a connection-mode network layer protocol. It actually consists of three protocols, one corresponding to each of the network, data link, and physical layers of the ISO Reference Model. This section will describe the highest of the three layers, the packet layer protocol (PLP). The X.25 PLP offers two types of virtual circuit service: virtual call and permanent virtual call. A permanent virtual call (PVC) is usually set up before an application needs to use it. There are no call set up or clear procedures necessary, in order for a network user to use a PVC. In contrast, a network user must always set up before, and clear after, using a virtual call (VC).

The standard for X.25 describes the DTE (data terminal equipment) sending packets to the DCE (data communication equipment) and vice-versa. The DTE can be thought of a computer with at least one layer above the network, data link, and physical layers. The DCE can be thought of a gateway computer, running just the network, data link, and physical layers.

The following table summarizes the X.25 packets/procedures:

CALL SETUP and CLEARING

DCE --> DTE	DTE --> DCE
incoming call	call request
call connected	call accepted

clear indication	clear request
DCE clear confirmation	DTE clear confirmation

DATA and INTERRUPT

DCE --> DTE	DTE --> DCE
DCE data	DTE data
DCE interrupt	DTE interrupt
DCE interrupt confirmation	DTE interrupt confirmation

FLOW CONTROL and RESET

DCE --> DTE	DTE --> DCE
DCE RR	DTE RR
DCE RNR	DTE RNR
	DTE REJ
reset indication	reset request
DCE reset confirmation	DTE reset confirmation

RESTART

DCE --> DTE	DTE --> DCE
restart confirmation	restart request
DCE restart confirmation	DTE restart confirmation

DIAGNOSTIC

DCE --> DTE	DTE --> DCE
diagnostic	

REGISTRATION

DCE --> DTE	DCE --> DTE
registration confirmation	registration request

Packet and Procedure Specifics

The following section will describe the function and format of the procedures and packets, as outlined in the table above. All packets contain the following "header" information. The first three octets of all X.25 packets contain:

- general format identifier
- logical channel group number
- logical channel number
- packet type identifier

Logical channel group number and Logical channel number

These two fields are used to denote one of 4095 possible logical channels between a DTE and its DCE(s). Length: 12 bits.

General format identifier

This field is used to denote packet sequencing options, confirmation receipt for data packets, and for denoting that a given packet contains a mixture of user data and control information. Bits 1 and 2 used to denote the type of sequencing for packets. There are two options here, modulo 8 and modulo 128. The former allows sequences of packet numbered 0 through 7. The latter allows sequences of packets numbered 0 through 127. The 3rd bit is known as the D bit. The setting of this bit indicates that either a confirmation receipt is desired, or that confirmation of packets is available. The 4th bit is also known as the Q bit. This bit, if used, denotes a mixture of user data and control information in the userdata field of the packet. Use of this field for such activities is explained in X.29. Total field size: 4 bits.

Packet type identifier

This field is used to denote the type of packet which follows the header. Length: 8 bits.

Call Setup and Clearing

Call set up and clearing procedures apply independently to each logical channel. The following packets are involved in the call setup phase:

- call request packet
- incoming call packet
- call accepted packet
- call connected packet

The call request packet is sent by the DTE to the DCE. The DTE selects a logical channel number, and this data, along with the called and calling DTE addresses, are passed in this packet.

The incoming call packet will be received by the DCE. The DCE will select a logical channel number to identify this call, possibly different from the one used by the originating DTE. This packet contains the called and calling DTE addresses.

The called DTE denotes acceptance of the incoming call by issuing a call accepted packet to its DCE. This packet contains the same logical channel number as the incoming call packet. After issuing this packet, the DTE is now ready to begin accepting data.

The call connected packet will be received by the calling DTE, if the called DTE has accepted the call request. At this point in time, the DCEs involved are able to identify this channel by the logical channel numbers selected by the called and calling DTEs. It is no longer necessary to send the destination DTE address in the data packets; all that is required is the logical channel number of the destination.

It is possible for a call request packet and incoming call packet to be transferred across the DTE/DCE interface with the same logical channel number. This situation is known as call collision. In order to resolve call collision, the following logic is used. The DCE will proceed with the call request packet, and reject the incoming call packet.

At anytime, the DTE may elect to close the virtual call and issue a clear request packet to its DCE. Upon receiving the clear request packet, the DCE will free the logical channel, and respond to the DTE with a local DCE clear confirmation packet. A clear request packet may also be issued by a DTE in response to an incoming call packet, as a way to reject the call request. The DCE then issues a clear request to the called DTE. The Called DTE's DCE receives the clear request, and sends a clear indication packet to its DTE. The DTE then responds with a DTE clear confirmation. At this time, the logical channel is again in the ready state, as it was before the call setup first occurred.

All call setup and clearing packets contain the generic header,as previously described. In

addition to the header, the 4 call setup packets contain the following fields:

Address block

This area is composed of several sub-fields, which includes the called DTE address length, calling DTE address length,called DTE address, and the calling DTE address. Length: variable, depending on DTE addresses.

Facility length field

This field is used to denote the length of the proceeding facilities field. Length: 8 bits.

Facilities

This field is present only when the DTE is using optional user facilities which require indications in the call request and incoming call packets. This field is limited by the facilities offered by the network. However, the maximum size allowed is 109 octets. Length: 0 to 109 octets.

Call user data field

This field, if present, is used to carry up to 128 octets of data. Note: this is not the usual method for carrying data over X.25. Length: 0 to 128 octets.

In addition to the header, the clear request and clear indication packets contain the following fields:

Clearing cause

This field is used to denote the reason for the clearing of the call. Some of the reasons for clearing include network congestion, number busy, and access barred. Length: 8 bits.

Diagnostic code

This field contains more specific information, in conjunction with the clearing cause, as

to why the call was cleared. Length: 8 bits.

Address block

See previous definition of this field.

Facility length

See previous definition of this field.

Facilities

See previous definition of this field.

Clear user data

This field may be optionally used to carry up to 128 octets of userdata. Length: 0 to 128 octets.

In addition to the header, the DTE and DCE clear confirmation packets contain the following fields:

Address block

See previous definition of this field.

Facility length

See previous definition of this field.

Facilities

See previous definition of this field.

Data and Interrupt

As is the case with call setup and clearing, the data and interrupt procedures apply independently to the logical channel in question. After call setup is accomplished, the logical channel is in a state to begin transferring data.

The standard maximum size of the user data field in a data packet is 128 octets. Other sizes may be negotiated on a DTE/DCE-wide basis, for a given permanent virtual call, or per virtual call. Once the maximum is agreed to any number of bits, up to the maximum, may be transferred across the DTE/DCE interface. It should be noted that some networks place a restriction on this, requiring that only integral number of octets be transferred.

At call setup time, the initiating DTE may elect to ask for end-to-end acknowledgement of data packets. This is negotiated and facilitated through the use of the "D" bit. The D bit is bit 7 in the GFI. In addition to end-to-end acknowledgment, it may be necessary to fragment user data into several network packets. This may be necessary to to packet size limitations on a given network. In this case, the "M" bit is used. The M bit works in conjunction with the D bit, and has an effect on the reassembly of segmented packets. When the initiating DTE sends packets with both the M and D bits set, then each packet arrives at the destination DTE in the same order as they were sent, and an acknowledgment is sent for each packet. If the M bit is set and D bit is 0, then the packet sequence will be reassembled by the network layer. The packet assembly will be complete when a packet arrives with the M bit set to 0. At this point, the packet is fully reassembled and will be passed to the network service user. If the D bit is set in the last packet, then an acknowledgement will be sent to the originating DTE.

The interrupt procedure allows a DTE to transmit data for which the normal flow control provisions do not apply. That is, an interrupt packet will be processed ahead of any data packets which the remote DTE may already have access to. To begin the interrupt procedure, the DTE transfers a DTE interrupt packet to its DCE. The remote DTE would receive a DCE interrupt, and acknowledge it with a DTE interrupt confirmation. The DCE associated with the originating DTE will receive the DTE interrupt confirmation and pass a DCE interrupt confirmation to the DTE. The originating DTE is not allowed

to send another DTE interrupt until it receives acknowledgment of the one sent. The DTE interrupt packet carries with it up to 32 octets of userdata.

In addition to the header, the DTE and DCE data packets contain the following additional field:

User data field

See previous definition of this field.

In addition to the header, the DTE and DCE interrupt packets contain the following additional field:

Interrupt user data

This field contains the interrupt user data.

The DTE and DCE interrupt confirmation packets contain only a header. No additional fields are required.

Flow Control and Reset

Flow control of data packets is controlled separately for each direction of a logical channel. Normally, flow control is accomplished through a sliding window mechanism. Each data packet contains a 3 bit send sequence number (P(S)), and a 3 bit receive sequence number (P(R)). A 7 bit sequencing scheme may also be optionally selected. With the 3 bit sequencing numbers, the default window size is 2 and can be set as high as 7. Data packets incoming on a logical channel contain a P(R) that is the number of the next packet expected to be received from the remote DTE.

When a packet is received with the D bit set to 1, then the DTE should return the corresponding P(R) as soon as possible. This may be accomplished via a data, receive ready (RR), or receive not ready (RNR) packet. If a DTE or DCE should receive a RNR packet, then it should cease sending data packets on that logical channel number. Data

transmission will resume when the DTE or DCE receives a RR packet or by initiation of a reset procedure.

In addition to the header, the DTE and DCE receive ready and receive not ready packets contain the following additional field:

Packet Receive Sequence Number

This field is used to denote the packet receive sequence number P(R). This means the remote DTE is ready (expects) to receive packet R next. Length: 8 bits.

Reset and Restart

These two facilites are used to recover from errors. The reset procedure serves to reset the virtual circuit and initialize the sequence numbers to 0 on both ends of the circuit. Any packets in transit may be lost, and it is the responsibility of higher layers to recover these packets. The reset procedure only applies where the virtual call is in a data transfer state. A DTE or DCE may initiate the reset procedure. To begin the reset procedure, the DTE transmits a reset request specifying the logical channel to be reset. The DCE will indicate a reset by issuing a reset indication, which includes the reason for the reset. The DTE upon receiving the reset indication will issue a DTE reset confirmation. The DCE will receive this DTE reset confirmation and send a DCE reset confirmation to its DTE. The logical channel is now in the reset state.

When a more serious error (such as network failure) occurs, a restart procedure is executed. This may be invoked by either the DTE or the DCE. The restart procedure clears all the virtual calls and resets all the permanent virtual circuits. The DTE may initiate a restart procedure by transmitting a restart request packet to its DCE. The DCE would then send to the remote DTE a DCE restart indication. The DTE would then transmit a DTE restart confirmation to the DCE, which would then send a DCE restart confirmation to its DTE.

In addition to the header, the reset request and reset indication packets contain the following fields:

Resetting cause

This field contains the reason for the reset. Length: 8 bits.

Diagnostic code

This field contains the diagnostic code and contains additional information with regard to the reset. Length: 8 bits.

The DTE and DCE reset confirmation packets contain a header and no additional fields.

In addition to the header, the restart request and restart indication packets contain the following fields:

Restarting cause

This field contains the reason for the restart. Length: 8 bits.

Diagnostic code

This field contains the diagnostic code and contains additional information with regard to the restart. Length: 8 bits.

The DTE and DCE restart confirmation packets contain a header and no additional fields.

Diagnostic

This procedure is used by some networks to indicate error conditions where the usual methods for error indication are inappropriate. The diagnostic packet from the DCE supplies information on error situations which are considered unrecoverable at the packet layer. The information given in the packet provides data for higher layers to possibly analyze and recover from the error. There is no corresponding confirmation packet expected from the DTE by the DCE.

In addition to the header, the diagnostic packet contains the following fields:

Diagnostic code

This field contains information on the error condition which resulted in the transmission of the diagnostic packet. Length: 8 bits.

Diagnostic explanation field

This field gives further information as to why the diagnostic packet was transmitted. The reasons can include reception of an erroneous packet by the DCE, or by a DCE timeout. Note: this service may not be available on every subnetwork. Length: 2 to 3 octets, depending on the cause of the error.

Now that the X.25 PLP has been described, the following section will explain how X.25 can be used to obtain the OSI network layer service. There are currently several versions of X.25 is use, namely the 1980 and 1984 versions. The mapping of the 1984 version of X.25 to the OSI network service will be explained here.

Provision of the OSI Network Layer Via X.25

The is very little which needs to be added to the 1984 version of X.25 in order to provide the OSI network service. What is required are guidelines for the use of X.25 in deriving this service. This is necessary in order for several groups to separately build an OSI network layer, and have it work with another group's OSI network layer. The specifications for this mapping include X.25 services and parameters, and their OSI network services and parameters counterparts. In addition, there is a detailed list of the optional X.25 facilities which are required.

Mapping X.25-to-OSI network services and parameters

The following table lists the OSI network service calls and parameters, and their counterparts in the X.25 PLP.

Network Connection Establishment Phase

OSI CONS PRIMITIVE	X.25 PACKET
N-Connect.request	Call Request
N-Connect.indication	Incoming Call
N-Connect.response	Call Accepted
N-Connect.Confirm	Call Connected

OSI CONS PARAMETER	X.25 FIELDS
Called Address	Called DTE Address
Calling Address	Calling DTE
Responding Address	Called DTE Address
Receipt Confirmation Selection	General Format Identifier
Expedited Data Selection	Expedited Data Selection Facility
QOS Parameter Set	Throughput Class Negotiation Facility
NS-User-data	Call and Called User Data

Network Connection Release Phase

OSI CONS PRIMITIVE	X.25 PACKET
N-Disconnect.request	Clear Request
N-Disconnect.indication	Clear Indication, Restart Indication, Clear Request

OSI CONS PARAMETERS	X.25 FIELDS
Originator and Reason	Cause and Diagnostic Code Fields
NS-User-data	Clear User Data
Responding Address	Called DTE Address

Data Transfer Service

OSI CONS PRIMITIVE	X.25 PACKET
N-Data.request	Data
N-Data.indication	Data

OSI CONS PARAMETERS	X.25 PACKET
N-User-data	User Data, M-bit
Confirmation Request	D-bit, P(S)

Expedited Data Transfer Service

OSI CONS PRIMITIVE	X.25 PACKET
N-Expedited-Data.Request	Interrupt
N-Expedited-Data.Indication	Interrupt

OSI CONS PARAMETERS	X.25 FIELDS
NS-User-data	Interrupt User Data

Reset Service

OSI CONS PRIMITIVES	X.25 PACKET
N-Reset.Request	Reset Request
N-Reset.Indication	Reset Indication, Reset Request
N-Reset.Response	none
N-Reset.Confirm	none

OSI CONS PARAMETERS	X.25 FIELDS
Originator and Reason	Cause and Diagnostic code fields

Sect. 4.9 - A Connectionless Network Protocol (OSI IP)

The protocol for providing the connectionless-mode network layer service is defined by ISO 8473 [33]. This protocol is also known as the ISO IP, i.e. the ISO Internet Protocol. The specification for this protocol is very much the same as the DARPA Internet Protocol. A brief description of the ISO version of this protocol will be made here for two reasons. One is that there are slight differences in the two protocols. The second is that part of the intent of this thesis is to acquaint the reader with the terminology and method for describing a given protocol.

The protocol described in ISO 8473 is designed to fulfill the role of a SNICP, as was previously described in a preceding section. This protocol functions to supply the OSI network service over a heterogeneous or homogeneous set of interconnected subnetworks. It is designed to accommodate SNDCPs and SNAcPs which provide all the functions necessary to support the connectionless-mode network services. The service provided by this protocol is specified in ISO 8348/AD1 [32].

The basic service offered by the network layer is the ability to send a network service data unit (NSDU) and get an indication that the data was delivered. Associated with this service are the parameters for source address, destination address, quality of service, and user data.

The underlying network service primitives exist to allow NSDUs to be passed from network entity to network entity. There is a request and indication pair, and associated with them the parameters for source address, destination address, quality of service, and user data.

There are some basic services,with regard to timers, that are required to from the local environment. They are: timer request, response, and cancel functions. The parameters associated with these parameters are timer value, name, and subscript. The timer value

represents the time needed to expire before a timer goes off. The name is the label associated with a given timer. The subscript indicates a unique timer, for a given name. The functions of the IP protocol are:

- PDU composition function
- PDU decomposition
- Header format analysis
- PSU lifetime control
- Forward PDU
- Segmentation
- Reassembly
- Discard PDU
- Error reporting
- PDU header error
- Padding
- Security
- Source routing
- Record route
- Quality of service maintenance
- Priority
- Congestion notification

Classification of Functions

Type 1: These functions must be supported.

Type 2: These functions may or may not be implemented. If an implementation does not support a type 2 function and the function is selected in a PDU, then the PDU must be discarded. An error report PDU may optionally be generated and sent back to the originating network entity.

Type 3: These functions may or may not be implemented. If an implementation does not support a type 3 function, and the function is selected, then the PDU is passed along as though the function was never selected. The PDU is not discarded.

PDU Structure

A PDU will always contain an integral number of octets, numbered from 1 to N. The bits in an octet are numbered from 1 to 8, where 1 represents the least significant bit. When a series of octets are used to represent a binary number, the lower numbered octets hold the bit in increasing significance.

Description of Functions

PDU Composition

This function is responsible for construction of a protocol data unit. The protocol control information is derived from local and current state data, and from the parameters associated with the N-unitdata request. Network protocol address information (NPAI) is obtained from the parameters associated with the NS-unit data request (i.e. the NS-source address and the NS-destination address). The data passed from the network service user becomes the data part of the PDU. To uniquely identify a given PDU, a data unit identifier is assigned by the originating NS user. This identifies a PDU from a specific source and to a specific destination. This is used to reassemble PDUs which get segmented (fragmented) by a subnetwork. The data unit identifier can also be used for the reporting of error conditions. If the non-segmenting subset protocol is in use, there is no need to use data unit identifiers, since reassembly is not needed.

PDU Lifetime Control

This function examines the lifetime field in the PDU header. The originator of the PDU selects a value for the lifetime field. As the PDU passes through a network entity and is

processed, its lifetime field is decremented by one. A value greater than one is decremented from it if the transit and processing time delay is greater than 500 milliseconds. If a lifetime value reaches zero, that PDU must be discarded. As a result, an error message PDU may be sent back to the originating network service user. Basically, this field is used to prevent PDUs from forever circulating through an internet.

PDU Decomposition

This function is responsible for removal and analysis of the PDU header. It then generates an N-unitdata indication. If the data part of the PDU is segmented, all the segments are collected before the N-unitdata indication is issued.

Header Analysis

This function has two main responsibilities: to determine whether the full or subset protocols is to be used, and to determine whether the PDU should be forwarded or not.

Route PDU

This function specifies the intermediate network entities that are to be traversed enroute to the destination. This function also specifies the underlying service that must be used. The route traversed by the PDUs is primarily influenced by the quality of service required. The results of this function are passed along to the forward PDU function.

Forward PDU

This function forwards PDUs using a subnetwork access protocol. It passes the address of the next system within the subnetwork-specific addressing domain, as well as the quality of service values.

Segmentation

This function is invoked when the PDU to be transmitted is larger than that which the underlying subnetwork supports. A PDU will be broken up into two or more "derived"

PDUs whenever segmenting is done. To identify these derived PDUs as being part of the original, the values of source and destination address, coupled with the data unit identifier are used to uniquely identify the pieces. The following PDU header fields are associated with the segmentation function:

- segment offset: identifies the octet at which this segment applies, with respect to the original PDU.
- segment length: the number of bytes in the derived PDU, which includes the header and data.
- more segments flag: this value is set to 1 if it this PDU does not contain the final octets for the original PDU.
- total length: the total length (header and data) of the original PDU.

There is a header field, segmentation permitted, which will be set to 1 if segmentation is permitted. If set to 0, segmentation is not allowed to take place.

Reassembly

This function is responsible for reassembling the derived PDUs back into the original PDU they were segmented from.

Discard PDU

This function discards a PDU for reasons such as: bad PDU checksum, PDU destination address is unknown or unreachable, PDU lifetime expired, and others.

Error Reporting

This function is invoked when it is necessary to send an error PDU back to the originating network entity. The header portion of the discarded PDU is sent back as the data portion of the error PDU. Error reporting concerning the discarded PDUs is suppressed if the data originating network entity set the error reporting flag to zero.

PDU Header Error Detection

This function computes a checksum over the entire PDU header. If the checksum does not verify, then it is discarded, and an error PDU may be generated. If the lifetime field (or any other) is modified during PDU processing, then a new checksum is computed and entered in the header checksum field. The use of header checksums is optional, and controlled by the source network entity.

Padding

This allows space to be reserved in the PDU header which does not support a specific function. An example use of this would be to align a data field on a computer word boundary.

Security

This function provides protection services, i.e. data origin authentication, data confidentiality, and data integrity for a single NSDU. This function is related to the protection from unauthorized access QOS parameter. It is realized by selection of the security parameter in the options part of the PDU header.

Source Routing

This function supports both complete and partial source routing. Source routing is the specification of the path to be traversed, from source to destination, of the NSDU. Complete source routing dictates that the list of of intermediate systems be traversed in the order specified, and that only those specified be traversed. Partial source routing is less restrictive. It mandates that all intermediate systems be traversed, but does not care if other, non-specified systems are visited in the process. In either case, if an intermediate system is not reachable, the PDU is discarded and an error PDU may be generated.

Record Route

This function supports both complete and partial route recording. Complete route

recording indicates that each intermediate network entity record its network entity title in a parameter in the options section of the PDU header. When this form of route recording is specified, reassembly of segmented PDUs is prohibited, unless all derived PDUs have same route path in their headers. For partial route recording, all intermediate network entity titles are listed in the PDU header options area. The difference between complete and partial route recording is that partial allows the reassembly of derived PDUs which may have used different paths in getting to the destination network entity. In this case, the route recorded in the reassembled PDU can contain the route from any one of its derived PDUs.

Quality of Service Maintenance

This function provides information to intermediate systems, which may be used to make routing decisions. It also may be used to request certain quality of service from subnetwork services.

Priority

This function provides a mechanism for PDUs with a numerically higher priority value, to be processed ahead of those with a lower priority value.

Congestion Notification

This function provides a means for an intermediate system to alert the destination NS user that congestion problems have been encountered. To indicate a congestion problem, an intermediate system would set to 1 the flag in the QOS parameters section of the PDU header.

PDU Structure

All PDUs shall contain the following parts in the following order:

- fixed part

- address part
- segmentation part
- options part
- data part

Fixed Part

The fixed part contains the following fields:

- network layer protocol identifier
- length indicator
- version identifier
- PDU lifetime
- flags
- type code
- PDU segment length
- PDU checksum

Network Layer Protocol Identifier

This field is set to the binary value of 1000 0001. This identifies the PDU as following the protocol described in ISO 8473. The value of this field is set to binary 0000 0000 to denote the inactive subset of the network layer protocol is in use. Length: 8 bits.

Length Indicator

This field denotes the header length,in octets. It may contain a maximum value of 254. Length: 8 bits.

Version Identifier

The value of this field is set to binary 0000 0001, to denote use of ISO 8473 version 1. Length: 8 bits.

PDU Lifetime

This field denotes the remaining lifetime value of a given PDU, in units of 500 milliseconds. Length: 8 bits.

Flags

This field is a composite of several fields:

- segmentation permitted
- more segments
- error report

Segmentation Permitted Flag

A value of 1 in this field indicates that PDU segmentation is permitted. This field cannot be changed after being sent by the originating network entity. Length: 1 bit.

More Segments Flag

This field denotes whether or not the data in a given PDU contains the last octet of data from the original PDU. When set to zero, the last octet of data in the PDU is the last octet of data from the original PDU. When set to one, there are segments outstanding. Length: 1 bit.

Error Reporting Flag

This field denotes whether or not a network entity should generate an error PDU when a PDU is discarded by a network entity. A value of one means an error PDU should be sent; a value of zero means no error PDU is sent. Length: 1 bit.

Type Code

This field denotes whether or not the field carries data or error information. Length: 5 bits.

PDU Segment Length

This field contains the length of the PDU, which includes both header and data, in octets. Length: 16 bits.

PDU Checksum

This field contains the value for the checksum computed on the PDU header. This field must be recomputed each time the PDU is processed and/or modified. Length: 16 bits.

The Address Part

The address part of the PDU header immediately follows the fixed part, and contains the following fields:

- destination address length
- destination address
- source address length
- source address

Destination Address Length

This field denotes the length of the destination address, in octets. Length: 8 bits.

Destination Address

This field holds the value for the destination address. This network service access point address is defined in ISO 8348/AD2 [32]. Length: variable, specified by the destination address length field.

Source Address Length

This field denotes the length of the source address, in octets. Length: 8 bits.

Source Address

This field holds the value for the source address. This network service access point

address is defined in ISO 8348/AD2 [32]. Length: variable, specified by the source address length field.

The Segmentation Part

The segmentation part of the PDU header is preceded by the address part. If the segmentation permitted flag is set to 1, then the segmentation part of the header must be present. It consists of the following fields:

- data unit identifier
- segment offset
- total length

Data Unit Identifier

This field is used to identify an initial PDU, and any of its subsequently derived PDUs. This identifier is used for PDU reassembly. Length: 16 bits.

Segment Offset

This field denotes the relative position of the data contained in the derived PDU, relative to the initial PDU. The value is in units of octets. The first derived PDU will contain an offset of zero; subsequent derived PDUs will contain a progressively higher, non-zero value. Length: 16 bits.

PDU Total Length

This field contains the value for the total length of the PDU. This includes both the header and data. This field does not change value, even for PDUs derived from it. Length: 16 bits.

The Options Part

This part of the PDU header is preceded by the Segment part. Options may appear in any order, but may not be duplicated. Options have the following general format:

- parameter code
- parameter length
- parameter value

Parameter Code

This field denotes the value of the option being selected. Length: 8 bits.

Parameter Length

This field denotes the length of the parameter value field, in octets. Length: 8 bits.

Parameter Value

The following parameter values are allowed in the parameter code field:

- padding
- security
- route recording
- quality of service maintenance

The Data Part

This part of the PDU header is immediately preceded by the Options part. The data part consists of the data passed in as the NS-user data parameter.

The next section will briefly describe the two PDUs which are specified by the network service:

- Data PDU (DT PDU)
- Error Report PDU (ER PDU)

Differences Between DT and ER PDUs

The DT PDU has the structure and fields as was previously described. The ER PDU has the same basic structure, with some minor differences:

- the flags portion of the fixed part is set to zero

- ER PDUs cannot be segmented, therefore they have no segment part
- the options part of the DT PDU are reflected by the options part of the ER PDU:
- priority and security parameter values are the same
- complete source routing is used to retrace the path taken by the DT PDU.
- padding, partial source routing, and record route are provided (if specified by the DT PDU) at the discretion of the error reporting network entity.
- the ER PDU contains two additional fields; one contains the reason for discarding the DT PDU, and the other contains the header of the discarded DT PDU.

Sect. 4.10 - Transport Layer Services

This section will explain the Transport layer services described by the ISO documents, namely ISO 8072 [26]. The main function of the of the transport layer is to achieve the service required by the higher layers, at a minimal cost [40], over a variety of subnetworks. It is often the case that the needs of the higher layers will not match what is available from the network layer. Parameters such as throughput, error rate, and end-to-end delay are used by the higher layers to convey to the transport layer what type of service is needed.

There are two types of services offered at the transport level, connection-mode and connectionless. The next section will describe the connection-mode service.

Connection-mode Service

The connection-mode transport service involves the following 3 phases:

- Connection Establishment
- Data Transfer
- Connection Release

The connection establishment phase involves the negotiation of facilities and parameters. The data transfer phase includes normal data, expedited data, and flow control.

Transport Connection-mode Primitives

The following list represents the service primitives and associated parameters, for a given phase of connection:

Transport Connection (TC) Establishment

- T-Connect.request (called address, calling address, expedited data option, quality of service, data)
- T-Connection.indication (called address, calling address, expedited data option,

quality of service, data)

- T-Connect.response (quality of service, responding address, expedited data option, data)
- T-Connection.confirm (quality of service, responding address, expedited data option, data)

Data Transfer

- T-Data.request (data)
- T-Data. indication (data)
- T-Expedited-Data.request (data)
- T-Expedited-Data.indication (data)

TC Release

- T-Disconnect.request (data)
- T-Disconnect.indication (disconnect reason, data)

The proceeding section will describe each of the 3 phases of the connection. Included will be a description of the service primitives used, and the parameters associated with the primitives.

Connection Establishment Phase

As is the case with the CONS, the connection mode transport service requires the establishment of a connection, prior to the data transfer phase. The following section details each parameter associated with connection establishment.

Called Address

This parameter conveys an address identifying the transport service access point (TSAP) to which the TC is desired. The TSAP is composed of a transport selector and a network address. It is this combination that serves to uniquely identify a user of the transport service.

Calling Address

This is defined exactly as the called address, except it is the address of the TC initiator.

Expedited Data Option

This parameter may have one of two values: enabled ot disabled. It should be noted that this parameter may be down-negotiated by the TS provider.

Quality of Service

The originating TS user first indicates the desired QOS, which may be altered by its TS provider. The remote TS user and/or provider may further alter the QOS parameter. The following items are contained in the QOS parameter set:

- TC Establishment Delay
- TC Establishment Failure Probability
- Throughput
- Transit Delay
- Residual Error Rate
- Transfer Failure Probability
- TC Release Delay
- TC Release Failure Probability
- TC Protection
- TC Priority

Data

Up to 32 octets of data may be sent in this parameter, although the session protocol does not use it.

Responding Address

This usually carries the same value as the called address in the TS-Connect.request/indication.

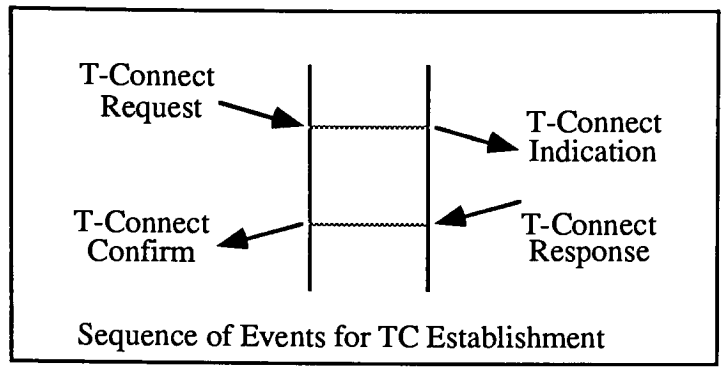
Sequence of Events for Connection Establishment

1. A T-Connect request is made by the TC user, to the TS provider. Parameters passed to the TS provider include the TSAP of the remote TS user, QOS, and expedited data option.
2. The TS provider receives the T-Connect request and determines whether it can satisfy it. If the request cannot be fulfilled, a T-Disconnect.indication is returned to the initiating TS user. One reason for the rejection of a request is that the minimum acceptable level of QOS could not be satisfied. If the request can be satisfied, it is sent to the remote TS user.
3. If the remote TS user cannot accept the connection, then it issues a T-Disconnect.request, which is received by the TS provider, which sends a T-Disconnect.indication to the initiating TS user. If the remote TS user accepts the connection, it sends back a T-Connect.response. The response will contain an indicator of whether the expedited data transfer option will be used. It could also contain a list of QOS parameters which it is willing to support. Note that the QOS parameters sent back may be lower than those suggested by the original TS-Connection.request.
4. When the originating TS provider receives the TS-Connect.response, it forwards it to the TS user as a T-Connect.indication. The TS provider may also downgrade the QOS parameters, to a level it chooses. This reduced level of QOS may not drop lower than the minimum acceptable level.

Assuming both sides of the TS user/provider agree on the connection request, a connection now exists between the TS users.

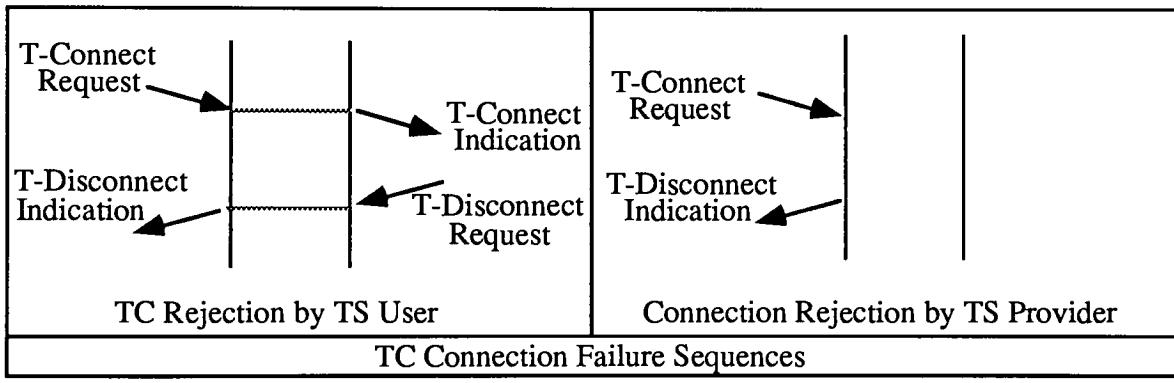
Sequence of TC Primitives

The following diagram shows the sequence of events for successful TC establishment:



Sequence of Events for TC Establishment

The connection establishment may fail due to the TS provider being unable to set up a connection with the remote TS user. The local TS provider may also reject the connection as well. This is shown in the following diagram:



TC Rejection by TS User

Connection Rejection by TS Provider

TC Connection Failure Sequences

Data Transfer Phase

Once a connection is established between two TS users, they are free to exchange data in either direction between themselves. There is no explicit confirmation of delivery. The ISO Transport standard [26] does not impose a limit on the amount of data which can be transferred with a single T-Data.request. It is left to the implementor to devise and plan for the size of the transport service data units (TSDUs). Expedited data, if negotiated for, allows for the transfer of data with a higher priority. There are no implied guarantees

regarding the delivery of expedited data versus regular data. It can be stated however, that regular data sent *after* expedited data will never overtake (and be delivered before) expedited data.

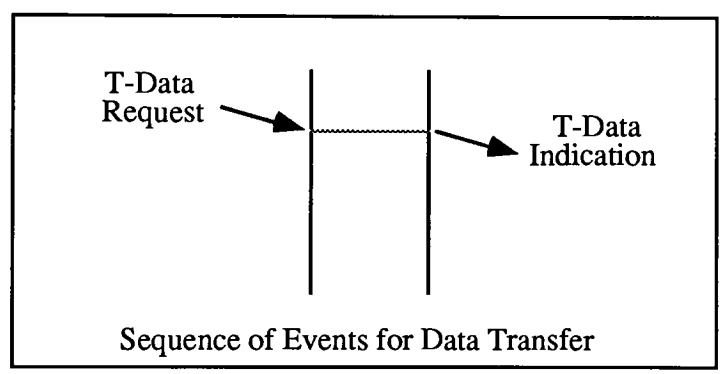
The following describes the single parameter associated with the data transfer phase.

Data

This parameter is used to convey the TS user's data. It must be at least 1 octet in length, and has no theoretical maximum. In reality, there will always be an upper limit on this maximum.

Sequence of Events for Data Transfer

The following diagram shows the sequence of events for transfer of data:



Sequence of Events for Data Transfer

The TC Release Phase

There are two parameters associated with the connection release phase: the T-Disconnect.request and the T-Disconnect.indication. The termination of a connection may be either user or provider initiated. When the connection release is invoked, data may be lost. It is the responsibility of the TS user to insure that there is no data in the transport connection when it is released. The TS user initiates the connection release process by issuing a T-Disconnect.request. The TS provider receives this request and

issues a T-Disconnect.indication to the remote TS user. The TS provider releases a connection by issuing a T-Disconnect.indication to both end TS users. The following describes the parameters associated with the TC release phase.

Data

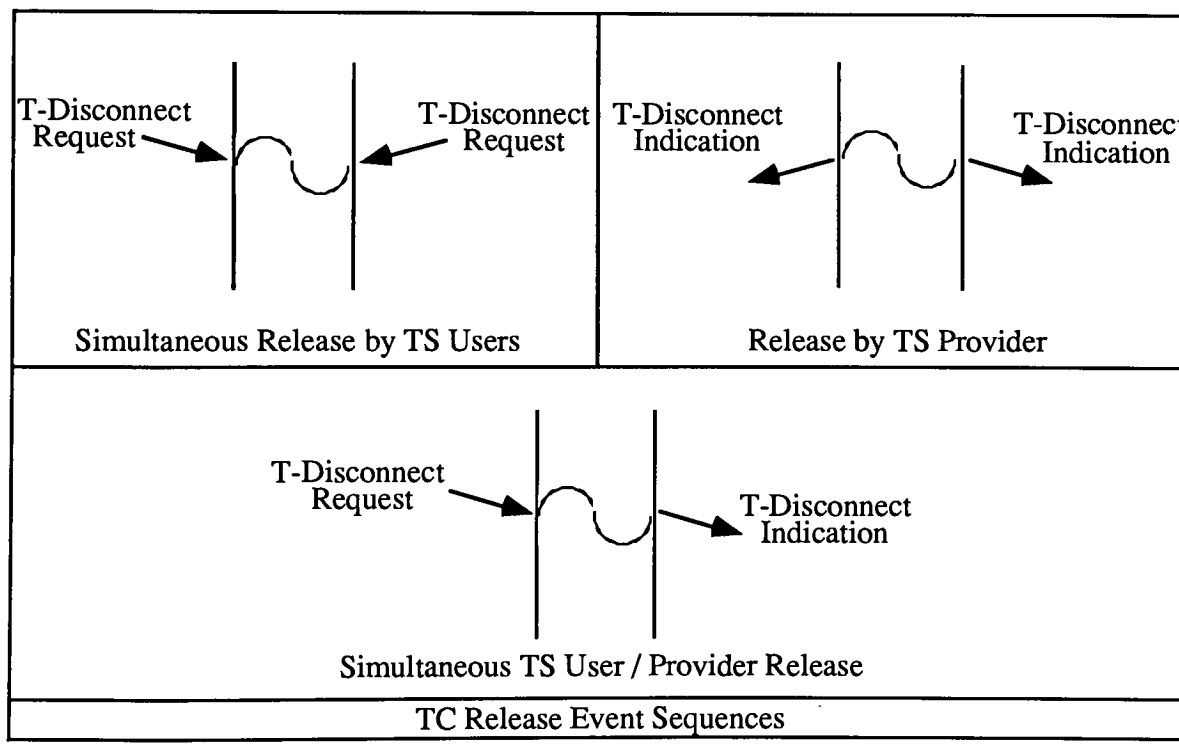
This parameter can be zero, or up to 64 octets in length.

Disconnect Reason

This parameter conveys information with regard as to why the connection was terminated. Valid reasons include QOS below minimum level, and lack of resources at the TS provider.

Sequence of Events for TC Release

The following diagram shows the sequence of events for TC Release:



TC Release Event Sequences

Sect. 4.11 - Transport Layer Protocols

This section will describe the OSI transport layer protocols for providing the OSI transport service. The OSI transport service standard [28] outlines the 5 protocols, any one of which may be used to provide the transport service. While each has the same calling interface, the complexity and sophistication varies drastically between them.

Part of the motivation behind the support of 5 transport protocols is the difference in the services offered by various subnets. The ISO, in standards document [28], describes three types of networks. Before discussing the transport protocols, the network types will be examined. The networks types are categorized as being type A, B, or C.

Type A

This subnet detects loss of data, but never duplicates, corrupts, or re-orders data. This subnet typically supports the CONS.

Type B

This subnet detects the loss of data (as in type A), but data loss is a more common occurrence. This subnet typically supports the CONS as well.

Type C

This subnet does not detect the loss of data. Moreover, data may be re-ordered, duplicated, or corrupted. This subnet typically supports the CLNS.

The following lists the 5 classes of transport protocol, along with its notable features.

Class 0 (TP0)

This is known as the simple class of transport protocol. It is responsible for setting up and terminating the TS connection. There is no error recovery, flow control, or multiplexing support. This protocol may be used only on type A subnets.

Class 1 (TP1)

This is known as the basic error recovery class of transport protocol. It is essentially the same as TP0, but can recover from network resets. This protocol may be run on type A or B subnets.

Class 2 (TP2)

This is known as the multiplexing class of transport protocol. It is essentially TP0 with support for multiplexing added. As such, it is suitable only for running on type A subnets.

Class 3 (TP3)

This is known as the error recovery and multiplexing class of transport protocol. It is essentially TP 1 with TP2 added. As such, it is suitable for running on class A or B subnets.

Class 4 (TP4)

This is known as the error recovery and detection class of transport protocol. It includes all the features of TP3, but it assumes the underlying subnet to be unreliable, and therefore takes steps necessary to make it reliable. This is the only transport protocol suitable for running on type C subnets. This is not to say that it would not work on type A or B subnets. In this case, there would be considerable redundancy in functionality.

The following table shows the three types of subnets and the transport protocols they will work with:

	transport protocol class				
subnet type	0	1	2	3	4
A	•	•	•	•	•
B		•		•	•
C					•

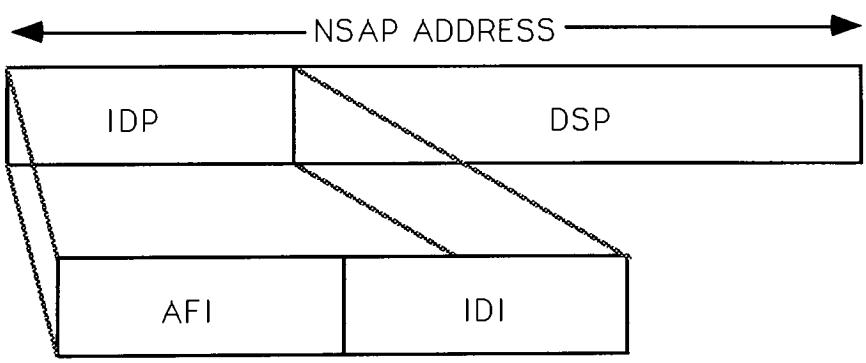
Subnet/Transport Protocol Compatibility Matrix

Sect. 4.12 - Analysis of OSI Internetworking Standards

This section will tie together the previous sections on the OSI protocols. The OSI protocols and services will be compared to the "ideal" internet discussed in Chapter 2, and these differences will be discussed.

Extensibility

To add a new node to an OSI subnetwork requires knowledge of who maintains the particular "domain" in question. Within a domain, administrations are chosen to handle addressing for given subdomains. The addressing authority is responsible for the semantics and syntax of the address; this part of the address may vary from subdomain to subdomain. Once a network address is obtained, other systems on the internet need to know how to get to the new system. Routing tables (static) or a directory service may be used to locate a path to the node. The problem with the OSI address space is that domains are allowed to specify the domain specific part of an address as they see fit. The following diagram and accompanying discussion describes the components of a NSAP address.



NSAP Address Structure

The authority and format identifier (AFI) is allocated by the ISO; the AFI determines the length of the initial domain identifier (IDI) and whether leading zeros have significance.

The AFI also determines the format of the domain specific part (DSP), i.e. whether it is binary coded or decimal. Probably the single largest problem in execution of the addressing scheme is the lack of address resolution protocols. That is, the method for determining a transmission address from the network address. This is predicted to incur heavy administrative burden [61]. Allocation of addresses (AFIs) is done through joint agreement of the ISO and CCITT [46]. Finally, addresses in OSI systems are not supposed to imply routing, but in reality many do [61].

Flexibility

The OSI protocols are layered such that they allow for new software and hardware technologies which may be developed. Like the Internet, the OSI protocols allow the selection of QOS parameters. These parameters include throughput, delay, error rate, resilience, transfer failure probability, access guards, priority, and cost. There are more options here than what the Internet protocols support. As will be discussed in a later section, the degree of flexibility with regard to the transport and network services may in fact be a disadvantage, or at least hamper internetworking to some degree. In theory though, the OSI protocols offer a high degree of flexibility.

Interoperability

There are currently few subnetworks which have the OSI networking protocols inplemented. Therefore, an implementation of the network layer interface must exist before interoperability can occur. The OSI network service currently has several standards documents which outline its use over a variety of heterogeneous subnets. This is in keeping with OSI's policy of openness. In theory, the OSI protocols should provide a high degree of interoperability. This is not, however, the case. First, the OSI protocols provide for both the connection and connectionless mode network services. This looks promising, but the political boundaries that this fosters become the difficult hurdle. The truth of the matter is that if a subnet supports the CONS, it is not likely to support the

CLNS as well. Then there is the matter of the transport service protocols. The OSI transport service is supported by no less than 5 protocols. They range from a very simple transport mechanism (class 0), to a full end-to-end reliability protocol (class 4). The OSI reference model prohibits the use of relays above the network layer, so if two end systems support non-compatible transport services, interoperability is not possible.

Robustness

The connection-oriented network protocols insure that data are reliably transferred at the network level. This is in direct opposition to the DARPA protocols, where reliability is built in at the transport level. Thus, once a network connection is established, a failure in any link between the end systems results in the need to open a new connection. Experience from the DARPA community has shown that subnets are inherently failure prone. The DARPA protocols address this issue of reliability at the transport level. It should be noted that because the OSI network service insures data are reliably transferred from intermediate system to intermediate system, that each intermediate system must run the procedures to insure this. This is seen by a disadvantage to some, that intermediate systems are required to accomplish these procedures. Again, with the DARPA protocols, the complexity is contained in the end systems only.

High Performance

Since the OSI protocols support both the CONS and CLNS, administrators have a choice. The same advantages outlined for the DARPA connectionless mode network protocol apply to the OSI CLNS as well. With regard to the OSI CONS, this service is usually a better performer than the CLNS for transferring large amounts of data in a given period of time. Also, since a connection is set up prior to data transfer, resources can be allocated in advance. This has the advantage of avoiding such things as buffer overflow and congestion, both of which contribute to low performance. The OSI CONS is still subject to similar performance degraders as network traffic and subnet bandwidth.

Data Integrity and Security

The OSI transport and network level services do not directly support data security; this is handled by the upper layers if it handled at all. Data integrity is maintained in a hop-by-hop basis, freeing the transport level from this task. Each intermediate system between end systems does not pass any data until it verifies that the data transferred has been sent and received correctly.

Transparency

Adding a new node or subnet to an OSI internet also has little impact on other already established nodes and subnets. The local subnet authority is usually the source for OSI network addresses. Once an address is obtained, it is only necessary to update routing tables. To this point in time, there are no routing protocols specified for OSI internetworks, thus requiring local protocols for handling routing data, and/or the use of static routing tables. If it is static tables that must be updated, then this could be somewhat less than the ideal imagined for transparency. Besides having to edit all required routing tables, a list or method for deriving such a list must be employed. Then the routing tables must be correctly edited.

The OSI CONS takes care of fragmenting and reassembling data as it is passed through intermediate systems, enroute to the destination end system.

Administrability

The ISO and CCITT are responsible for allocating initial domain identifier (IDI) part of the OSI network addresses. The domain specific part (DSP) is left to the domain authority to allocate. If a group or groups wishes to organize their own internet, then they are free to assign network addresses as they see fit. However, OSI guidelines for the allocation of the IDI must be followed if the private internet is to ever attach to other OSI internets, due to the need for OSI network address uniqueness. As far as other

administrative tools go, they are currently being defined by international committee.

Traffic Accountability

The OSI committees are working on network level accounting, which would cover the following:

- number of packets sent/received
- number of octets sent/received
- time of day
- duration of connection

There is also consideration of where the accounting information should be sent. The OSI CONS has a good chance, by its very design, of capturing the traffic data necessary for billing purposes.

Chapter 5 - The DARPA Internet

This section will contain a general overview of the DARPA Internet. Included will be a listing of the various layers, and the protocols and services offered there. Specifically, the following protocols will be examined:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Transmission Control Protocol (TCP)
- Gateway-to-Gateway Protocol (GGP)
- Exterior Gateway Protocol (EGP)
- Interior Gateway Protocol (IGP)

Finally, this chapter will close with an analysis of the DARPA Internet Protocols. The Protocols will be compared to the features which have been previously identified for an ideal internetwork.

Sect. 5.1 - The Internet Services and Protocols

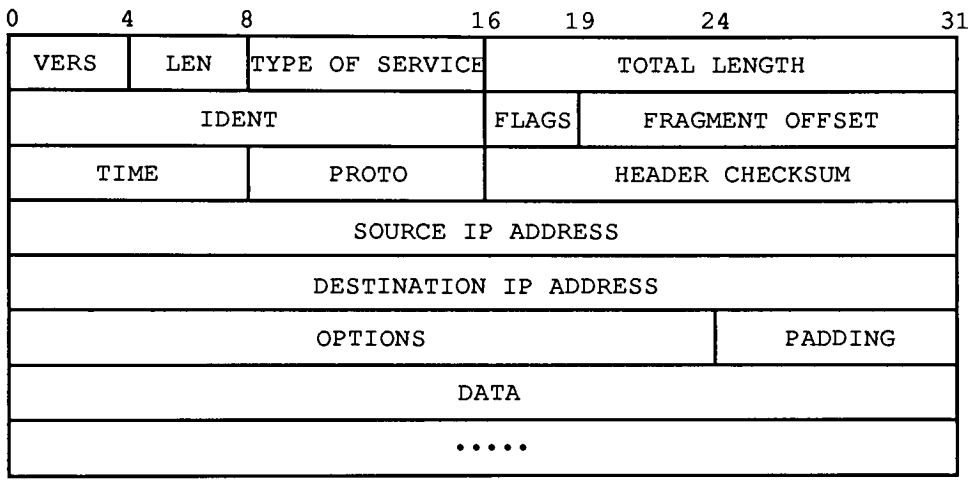
This section will describe how the DARPA Internet functions. Included will be a discussion of the Internet Protocol (IP), the Internet Control Message Protocol (ICMP), the Transmission Control Protocol (TCP), the Gateway to Gateway Protocol (GGP), the Exterior Gateway Protocol (EGP), and the Interior Gateway Protocol (IGP).

Before discussing the specifics of the TCP/IP protocols, some background and design philosophy is in order. The Internet can be described as a collection of various networks including NSFnet, NYSERnet, university and research networks, as well as several military networks [7]. Research on internetworking has been an ongoing effort by the DARPA, for approximately the last 20 years [8]. Systems employing the TCP and IP protocols have found use in both civilian and military applications. As experience with the protocols grew, the protocols themselves evolved to their current configuration. For example, the division of services between the TCP and IP levels, as well as the notion of a connectionless service, were not part of the original design [13]. One of the primary thrusts of the DARPA internet was to connect and multiplex existing subnetworks. This interconnection of subnetworks would add value to the system as a whole. Resources could be shared from one subnetwork to another, which has implications towards increased use and economies. One of the basic building blocks of the Internet was the concept of packet switching. This is the method by which data is moved, via gateway(s), to a destination. The TCP/IP protocols are designed to work as a highly reliable method for transporting data between nodes on the same networks, as well as nodes residing on logically or physically different networks. These protocols strongly imply a packet-switched architecture.

Sect. 5.2 - The Internet Protocol (IP)

This section will describe the IP protocol. The IP is responsible for routing (addressing) and fragmentation/reassembly of datagrams. A datagram is a unit of data, and this is what is handled by the TCP and IP protocols. The IP module is passed a destination address and a datagram to send there. It then attaches to the datagram an IP header. The contents of the header are fully described in [49], and will be summarized here.

The following diagram illustrates the structure of an IP datagram:



IP Datagram Structure

The following list contains IP header field names and descriptions.

Version

The Version field denotes the format type of the internet header. The format described here is version 4. Length: 4 bits.

IHL (Internet Header Length)

This field denotes the length of the internet header, in 32 bit quantities. The minimum legal value for this field is 5. Length: 4 bits.

Type of Service

This field provides a mechanism to request a given form of service be utilized when sending the datagram through the Internet. A datagram can specify between normal delay, throughput, and reliability, or low delay, high throughput, and high reliability. Length: 8 bits.

Total Length

This field denotes the total length of the datagrams, as measured in octets, and includes the header. This field, then, allows the total length of the datagram to be 65,535 (65,535 8 bit entities). It is generally agreed that all IP modules are able to accept datagrams up to 576 octets in size. Further, it is recommended that datagrams larger than this not be sent, unless the destination is specifically able to handle it. The size 576 was selected as it lends itself nicely to sending 512 octets of data along with a header of 64 octets. Length: 16 bits.

Identification

This field is used to denote the identification of a given datagram, n cases where the datagram has been fragmented and needs reassembly at the destination. Length: 16 bits.

Flags

This field contains the flags which tell a given IP module whether the datagram can be fragmented, has been fragmented, and/or represents the last fragment of a datagram. Length: 3 bits.

Fragment Offset

This field denotes the location within the datagram that this fragment belongs. The unit of measure is the octet. The first fragment of a datagram always has offset equal to zero. Length: 13 bits.

Time to Live

This field denotes the maximum time a given datagram is allowed to "live" on the internet. A datagram with a Time to Live equal to zero must be destroyed. The intent here is to delete all undeliverable datagrams, or to limit the possibility of a datagram looping through the Internet forever. Although this field is thought of as a measure of time, it value is usually decremented as it passes through intermediate gateway(s). Length: 8 bits.

Protocol

This field is used to denote the higher level protocol which is used in the data portion of the datagram. The values for this field are described in "Assigned Numbers", RFC 790 [48]. Length: 8 bits.

Header Checksum

This field contains the value of the checksum for the IP header only. Due to the changing nature of some fields (i.e. Time to Live), the checksum is recomputed at each place the header is processed. Length: 16 bits.

Source Address

This field contains the address of the source of the datagram. Length: 32 bits.

Destination Address

This field contains the address of the destination of the datagram. Length: 32 bits.

Options

This area is optional and need not appear in a datagram. It is only required that a given IP module handle the options area. There exist two formats for the options area: a single octet of option type, or an option-type octet, followed by an option-length octet, followed by the option octets themselves. The option-length octet contains the octet count of the entire options area.

The option-type octet has the following three fields:

- 1 bit copied flag
- 2 bits option class
- 5 bits option number

The copied flag bit signifies whether the option data should be copied to fragments (not copied => 0).

The option classes are as follows:

- 0 - control
- 1 - reserved for future use
- 2 - debugging and measurement
- 3 - reserved for future use

The options field(s) can provide for the following data:

- end of options list
- no operation
- security
- loose source and record route
- strict source and record route
- record route
- stream identifier
- Internet timestamp
- padding

The next section will elaborate on how IP accomplishes its addressing and fragmentation functions.

Addressing in IP

The source and destination addresses within a given datagram can be interpreted in one of four ways:

class	high order bits	format
a	0	7 bit network / 24 bit host
b	10	14 bit network / 16 bit host
c	110	21 bit network / 8 bit host
d	1110	extended addressing mode

By allowing for the variability of the Internet address fields, a range of configurations is supported. Configurations supported range from a small number of networks with a large number of hosts, to a large number of networks and a small number of hosts. Currently the extended addressing mode is undefined.

Fragmentation and Reassembly

The internet identification field, source address, destination address, and the protocol field are all used to identify datagrams which have been fragmented. To signify that a packet has more pieces (datagrams), the more fragments flag in the header is set. In order to identify the position of a given fragment in the original datagram, the fragment offset field is used. It signifies the relative position of a fragment in the original datagram. This field is in units of 8 octets.

Sect. 5.3 - The Internet Control Message Protocol (ICMP)

This section will describe the ICMP protocol. This protocol is fully described in [50]. ICMP is used between gateways in the Internet, to communicate information regarding the status of datagram transfers. The IP protocol is not assumed to be 100% reliable and it is the purpose of ICMP to provide information on problems that may occur. ICMP does not report any errors it may encounter sending ICMP messages, nor does it guarantee receipt of an ICMP message for a given communication error. If reliability is desired, then it is up to the higher level protocols (such as TCP) to provide it.

An ICMP message is sent using a standard IP header. In the data part of the datagram resides information as to the type of ICMP message and the subsequent data associated with that particular message type.

ICMP Message Types

- Destination Unreachable
- Time Exceeded
- Parameter Problem
- Source Quench
- Redirect
- Echo Request/Reply
- Timestamp Request/Reply
- Information Request/Reply

ICMP Message Format

All ICMP messages have the same basic format. Each has a type field (8 bits) which denotes one of the eleven types of ICMP messages. The code field (8 Bits) contains information specific to the type message sent, and varies based on message type. The checksum field (16 bits) represents the one's complement of the ICMP message, starting

with the type field. The next area consists of a field 32 bits wide, which also contains data specific to a given message type. Some messages use this area, others leave it blank (zero). The next area contains the first 64 bits of the IP header of the datagram found to be in error. The data in the header is used to associate the message with the sending process at the source. ICMP assumes that the port numbers (if used) in the higher level protocol (i.e. TCP) will be contained in the first 64 bits of the IP datagram data area.

ICMP Message Descriptions

Destination Unreachable

Reasons for sending this message back to the originator of the source datagram include the determination that:

- a subnetwork is unreachable from a given gateway
- a host is unreachable from a given gateway
- the IP module cannot deliver a datagram due to the port being inactive
- a datagram needs to be fragmented to pass through a subnetwork, but the do not fragment flag is set

Time Exceeded

If a gateway receives a datagram which has a time to live field equal to zero, it discards the datagram and sends a time exceeded message to the source. Another reason for sending this message is that the destination timed out while waiting for all the fragments for a given datagram to arrive.

Parameter Problem

This message is sent to the source if a parameter in the IP header is bad. A value of 1 in the code field of the ICMP header indicates that the pointer field contains the octet number of the bad parameter. This message is sent if the datagram is discarded.

Source Quench

This message is sent to the source if a gateway or destination cannot buffer the incoming datagram. This message may be sent at the time buffer space is totally consumed, or prior to total consumption. The idea of this message is to throttle back a source that is forwarding datagrams too quickly to the destination.

Redirect

This message is sent if a shorter path between a source and intermediate gateway is known. For example, if a source sends a datagram to a gateway on the same network, which would in turn forward it to yet another gateway on the same network, a redirect message would be sent by the first gateway.

This message instructs the source that a shorter route exists, and forwards the Internet address of that gateway. If source routing options are being used, then the path to the destination is blindly followed and no redirect messages are generated.

Echo Request/Reply

This request/reply pair is used to echo messages between a source and destination. The receiver of an echo request merely switches the source and destination addresses in the ICMP header, recomputes the checksum, and sends the datagram back to the source. A sequence number is provided so that the sender of the echo request can match it to a corresponding reply.

Timestamp Request/Reply

For this message, a timestamp is inserted in the data area, just prior to the source sending the datagram to the destination. At the destination, a timestamp is added at the time the datagram is received, and again just prior to it being sent back to the source. The destination reverses the source and destination addresses in the ICMP header and forwards the datagram back to the source. All timestamps are assumed to be valued in

seconds since midnight, Universal Time (UT). If this format is not being used, the high order bit of the timestamp field is set.

Information Request/Reply

This message comprises a method for a source host to determine the number of the subnetwork to which it attaches. The destination field of this message is set to zero, which denotes the current subnetwork. The reply sent back by the replying IP contains a fully specified address.

Sect. 5.4 - The Transmission Control Protocol (TCP)

This section will provide an overview of how the TCP works. RFC 793, "The Transmission Control Protocol" [51], describes the TCP protocol in detail. The TCP is designed to provide a reliable, end-to-end (process to process) protocol for packet switched networks. It is assumed that the host processes may be located on different, but connected subnetworks. The TCP makes no assumptions regarding the reliability of the protocols or subnetworks using these over which the data may travel. The reliability is built into the TCP itself. The TCP is designed to sit above and interface to the IP. As was described earlier, IP is the mechanism by which datagrams are communicated through the various subnetworks to a destination. Above the TCP lies the user application program or protocol.

In order to provide reliable communications, the TCP must address the following issues: basic data transfer, reliability, flow control, multiplexing, connections, and precedence and security.

Basic Data Transfer

The TCP allows the transfer of a stream of octets between two or more TCP modules. The actual size of the segments sent from the module is dependent on a size, agreed upon size at connection open time.

Reliability

In order to detect and recover from lost, duplicated, out of order, or damaged data, the TCP attaches sequence numbers to each octet transmitted. It is expected that each TCP segment will be acknowledged by the receiving module, along with the sequence number (first octet in the segment). Segments are ordered at the receiving end according to the sequence numbers associated with each segment. If an acknowledgement is not received by the transmitting module within a specified timeout period, that segment will be

retransmitted. To detect a damaged segment, a checksum is included. Damaged segments are discarded, prompting a retransmission from the transmitting TCP module. The TCP will function in the face of communication system errors, as long as the internet does not become partitioned. That is, as long as there exists a path somewhere in the internet from source and destination, the TCP will be able to communicate the data.

Flow Control

Flow control is provided via a "window" mechanism. As segments are received, the receiver acknowledges with the window information indicating the range of sequence numbers which the sender is allowed to send.

Multiplexing

The TCP allows for multiplexing; this permits a module running on a single host to communicate with several modules and or processes on other hosts. The TCP associates a port number with each process that accesses it. The subnetwork and host addresses along with the port number are known as a socket. Two sockets serve to identify a given connection uniquely. Usually, it is the responsibility of the host to associate port numbers with processes. The exception to this rule is the notion of well-known ports. Well-known ports represent services such as mail, remote login, and naming. These services have, in essence, hard-coded port numbers for all hosts. The port numbers of other, mainly user written, application services involve a more dynamic port to process association scheme. For example, an application may register with a directory service to advertise both a service, as well as as port number for the service.

Connections

A TCP connection between two modules causes resources at each to be allocated. These resources include memory to store status information such as: window size, sequence numbers, and the socket data. This information is stored in a Transmission Control Block

(TCB). In order for two TCPs to communicate, the connection must first be initialized with the above information. At the close of the connection, the information is discarded and the resources deallocated.

Precedence and Security

TCP allows for the user to declare the relative amounts of security and precedence required for the communication. Defaults are used if no values are specified.

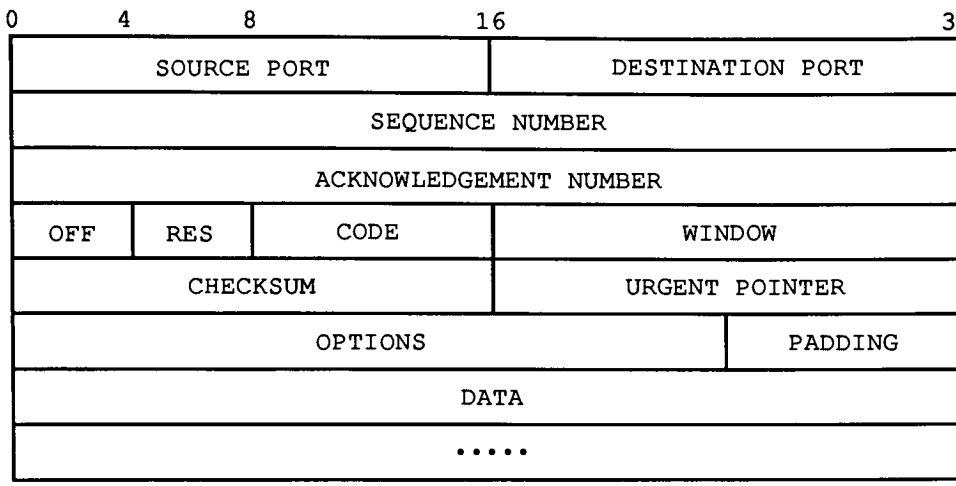
Opening a TCP Connection

A process may initiate the opening of a connection in one of two ways. The process may either make an active or a passive open. An active open is a request to the local TCP to make a connection with another specific module (i.e. the TCP running on a specific machine), or possibly any system running the TCP which offers a specific service. A passive open allows a process the ability to listen and possibly accept requests from other modules (which would be initiating an active open call). If the module making the passive open didn't care who connected, it would provide an unspecified foreign socket in the open call. If it wanted to limit who could connect, then it would specify a specific socket in the open call.

TCP Header Format

The following section describes the format and meaning of each field in the TCP header. TCP segments are communicated as internet datagrams. The TCP header and data segment are packaged by the IP and sent through the internet. The separation of TCP information from IP information allows the ability of protocols other than TCP to utilize IP.

The following diagram illustrates the TCP packet format:



TCP Packet Format

TCP Header Fields

Source Port

This represents the port number of the source TCP. Length: 16 bits.

Destination Port

This represents the port number of the destination TCP. Length: 16 bits.

Sequence Number

The sequence number represents the octet number of the first octet in the data segment. Length: 32 bits.

Acknowledgment Number

This fields contains the value of the next sequence number the sender of the segment is expecting to receive. The information in this field is examined only if the ACK control bit is set. Length: 32 bits.

Data Offset

This fields contains the number of 32 bit words in the TCP header. Immediately following the header is the data segment, so this field in effect is a pointer to the data segment. Length: 4 bits.

Reserved

This field is reserved for future use. It must contain only the value zero. Length: 6 bits.

Control bits

This field designates the following control information:

- URG - urgent pointer field in use
- ACK - acknowledgment field in use
- PSH - push function to be invoked
- RST - reset the connection
- SYN - synchronize the sequence numbers
- FIN - no more data to send

Length: 6 bits.

Window

This field denotes the number of octets, starting from the one in the acknowledgment field, that the receiver is willing to accept. Length: 16 bits.

Checksum

This field contains the checksum, computed on the TCP header and data segment. If the header and data are not an even number of 16 bit words long, then it is padded with an extra 16 bit (all zero) word. This word is not transmitted. As the checksum is being computed, the checksum field is set to zero. After the checksum is computed, the value is placed into the checksum field. Length: 16 bits.

Urgent Pointer

This field may contain a value which points to the sequence number of the octet following the urgent data. The value in this field is valid only when the urgent data bit is set. Length: 16 bits.

Options

The options area follows the end of the TCP header, but comes before the data segment. A given TCP is not required to use this area, but must be able to accommodate other TCPs which may. There are two classes of option formats. One format is simply an octet of option data. The other format contains an octet of option-kind, an octet of option length, followed by the actual option data. Currently there are three kinds of options:

- 0 - signifies end of the option list
- 1 no operation (used to align the next option on a word boundary)
- 2 - maximum segment size

Length: 8 - 16 bits.

Padding

This field is used to insure the TCP header ends on a 32 bit boundary. It always has a value of zero. Length: 0 - 24 bits.

Sect. 5.5 - The Gateway-to-Gateway Protocol (GGP)

This section will describe the GGP protocol. The GGP is fully described in [52]. The GGP is used by the core gateways in the Internet. These gateways are distributed throughout the U.S. and are centrally administered by the Network Operations Center (INOC). All other gateways on the Internet are classed as non-core gateways, and do not use GGP.

The GGP is used to ascertain connectivity information and distribute it to other networks and neighbor gateways. From this data comes the routing information used by the other core gateways.

A gateway will advertise its connectivity to other gateways via routing update messages. These update messages list the networks which are reachable by the gateway, and a hop count to get to each one.

A gateway which connects directly to a network is considered to have a hop count equal to zero. In general, a gateway which has to go through N gateways to reach a network is considered to have a hop count of N.

When are routing updates sent out? When

- a network interface changes state
- a neighbor gateway changes state
- a routing update is received from a neighbor, and it differs from the last update

To compute a routing update, a gateway will examine its routing table. It will extract the shortest route to a given network and pass that information on to its neighbors. As a gateway receives updates from its neighbors, it constantly updates its own routing tables, adding any new network numbers it receives. If a routing update is received from an apparently new neighbor (i.e. a neighbor address not found in the neighbor table), then

the new neighbor address is added. The new routing information from that neighbor will not be used until the GGP polling mechanism can be used to verify that the new neighbor is up. Thus in order for a new neighbor gateway to join the core and use GGP, it must be given a list of gateways it is to be neighbors with. The neighbors in the list will update their neighbor tables as routing updates are received from the new neighbor.

Besides routing updates, the GGP includes several other message types. An echo request/reply pair is defined, in order that a gateway can poll its neighbor(s) at regular intervals. Neighbors are polled at regular intervals to determine whether they are up and able to route datagrams. An acknowledgement reply message is defined in order that a neighbor, upon receipt of a routing update, can acknowledge a routing update which it received. A status message also exists, which serves as a method for a gateway to test its neighbor connection(s), to determine whether they are operational.

Advantages/Disadvantages

It is desirable to have only a small number of gateways knowledgeable about the entire topology of the Internet. This directly affects the amount of routing data that must traverse the Internet, and aspires to minimize it. Unfortunately, the routing tables of all the core gateways that implement the GGP must reach an equilibrium, or converge [14]. If not, routing updates will continually be exchanged between all neighbors, for no good reason.

Sect. 5.6 - The Exterior Gateway Protocol (EGP)

This section will describe the EGP. The EGP is fully described in [54]. The GGP is used to exchange routing data between the core gateways of the Internet. Somehow, routing information must be exchanged between these core gateways and the other, non-core, gateways connected in/directly to the Intemet. The protocol used for this purpose is EGP.

The main reason for using EGP and not GGP for this purpose is mainly due to the fact that GGP does not work well with a large number of gateways. Specifically, these are some of the problems [14]:

- overhead of routing data becomes too large. Routing data exchanged between gateways becomes extremely large, and whenever a topology change occurs, this data must be forwarded between all the gateways participating in the protocol.
- too many (different) gateways participating in the same routing algorithm makes error detection and maintenance nearly impossible.
- imposing the same routing algorithm for all gateways is in no one's favor; it makes any changes too widespread and thus affects too many machines/administrations.

Autonomous Systems

Because the structure of subnetworks at a given site can be arbitrarily complex, the notion of an autonomous system was developed. An autonomous system is a collection of networks and gateways under a single administration authority [14,56]. Within an autonomous system there is freedom to choose the method for obtaining and maintaining data regarding internal routing and connections. When two autonomous systems wish to share reachability data, they do so via EGP. It is essential that every autonomous system pass its reachability data to the core gateways on the Intemet.

The notion of neighbors is an important one in the EGP. Two gateways are neighbors of each other if they interface to the same subnetwork. They are interior neighbors if the

subnetwork to which they interface is wholly contained in the autonomous system. They are considered exterior neighbors if the subnetwork they interface to connects the autonomous systems to which each belong. An example of exterior neighbors is a subnetwork connection between a core gateway and a gateway contained within a separate autonomous system.

The exterior neighbors use the EGP to advertise reachability information for subnetworks within their autonomous system, ONLY. This is a key point Also, since EGP messages are only exchanged between exterior neighbors, messages are exchanged directly between the neighbors (i.e. there are no intervening gateways to pass through).
There are three parts to the EGP:

- neighbor acquisition
- neighbor reachability
- network reachability

Neighbor Acquisition

In order to begin exchanging routing data, two exterior gateways must first become direct neighbors via neighbor acquisition. It is assumed that a gateway knows in advance with which gateways it wishes to become neighbors. A neighbor acquisition request message is sent. If successful, the receiving gateway will send a neighbor acquisition reply. Included in the reply is a time interval, suggesting a polling interval for the neighbor reachability messages to be exchanged.

Neighbor Reachability

At least one of the direct neighbors will send "hello" messages to the gateway it is direct neighbors with. It will then wait to receive "I heard you" message in reply. This follows a K out of N methodology; for N hello messages sent to a given neighbor, the sender expects to receive K I heard you reply messages. A gateway can operate in active mode

or passive mode. Active mode means the gateway sends hello's and expects replies, to determine neighbor reachability. Passive mode means the gateway depends on direct neighbors which use active mode. That is, a passive gateway listens for "hello" messages to determine the reachability of a given neighbor. Passive gateways also reply to hello messages with "I heard you" replies.

Network Reachability

To distribute information regarding network reachability, a gateway will send routing update messages. It is important to note that EGP restricts non-core gateways to advertising networks reachable only within the autonomous systems of which it is a part. This rule is intended to restrict autonomous systems in advertising reachability information only to subnetworks within the autonomous system.

The Network Reachability (NR) message contains a list of gateways in the autonomous system, each followed by the networks for which this gateway is an appropriate first hop. In addition to this, the NR message also states the distance a given network is from the specified gateway. The EGP does not impose a specific measure of distance. The distance is an index which is only valid in comparing distances within a given autonomous system. NR messages can originate from either of two exterior neighbors. This means that a core gateway will share reachability information with a non-core gateway, and vice-versa.

Sect. 5.7 - The Interior Gateway Protocol (IGP)

This section will describe the IGP. IGP is described in [14]. Autonomous systems use EGP to communicate reachability information between themselves. In order for the exterior gateway and its associated interior gateways to exchange routing information, an IGP is employed. Whereas exterior gateways use a standard version of EGP, there is no single standard for the IGP. Three forms of IGP are currently in use. GGP, Routing Information Protocol (RIP), and HELLO are commonly used.

GGP has been previously described, and is the IGP used by the core gateways.

The most widely used IGP is RIP [14]. It is also known by the name of the program which implements it: *routed.* The popularity of RIP is due mainly to the fact that it has been distributed with the 4.X versions of Berkeley UNIX. RIP works by having each gateway which employs it broadcast its current routing database occasionally. The number of hops to each of the gateway's reachable subnetworks is kept in the routing database. A hop count of 15 denotes infinity and implies that the autonomous systems where it is possible to have > 15 hops to get out are not a good candidate for implementing RIP. Another disadvantage of RIP is that routing update messages propagate slowly, and this can lead to slow convergence. Slow convergence means that a significant amount of time is required for several gateways to realize a subnetwork link is down, and therefore unreachable.

The HELLO Protocol

HELLO is an interior gateway protocol, which uses metrics based on subnetwork delay. To accomplish this, the protocol must also synchronize the clocks on the gateways within the autonomous system. When a gateway exchanges routing data, it adds a timestamp just before sending out the packet. When the destination gateway receives the request, it places a timestamp in the datagram before sending. The source, upon receipt of the reply

datagram, subtracts its notion of that gateway's clock, in order to calculate a subnetwork transmission time. In this way, gateways can calculate the shortest time path to a given subnetwork, independent of the number of hops.

Sect. 5.8 - Analysis of the DARPA Internet

This section will analyze the DARPA Internet. It will include a comparison of the functionality offered by the DARPA protocols and that offered by the ideal internetwork.

The following section will analyze the DARPA internetworking protocols according to the requirements outlined in a previous section.

Extensibility

To add a new node to an existing subnet, one must first obtain an IP address. The local administrators are responsible for allocating new IP addresses. When a subnetwork is first attached to the Internet, the Network Information Center (NIC) at SRI International assigns it a network address, which is the prefix contained in all IP addresses in that particular subnet. If there is no desire to ever connect to the Internet, the the local administrators can assign IP addresses as they see fit. It should be noted that while it is easy to add new nodes to the Internet, the address space is limited by the fact that only 32 bits have been allocated for all IP addresses. An IP address strongly implies the route a packet must take in reaching its destination. Also, where the OSI protocols lack the address resolution protocols to determine a transmission address from a network address, the Internet protocols include an address resolution protocol (ARP) [53], designed expressly for this purpose.

Flexibility

The DARPA Internet has evolved over the years, changing as new algorithms were tried and as hardware advances were made. As has been previously shown, the Internet protocols have a layered architecture, which makes changes to specific functionality independent of other functions. For example, if changes are required in the way routing tables are handled, it is not necessary for the routing algorithms to necessarily change, as a result.

In terms of other flexibility, the Internet protocols allow the specification of quality of service parameters. Specifically, the IP accepts the parameters which denote normal delay, throughput, and reliability, or low delay, high throughput and reliability. The D, T, and R bits in the IP header are used to specify the delay, throughput, and reliability, but there are no guarantees that all gateways will implement these quality of service parameters. It is unclear how the layers above the IP layer directly access the type of service parameters.

Interoperability

The TCP/IP protocols have demonstrated the ability to interface to a wide variety of hosts and subnets. The IP expects very little in the way of facilities from a subnetwork. This results in the ability of it to run over a wide variety of subnetwork-specific protocols, such as X.25. The IP has also been implemented above token ring, broadcast, serial lines, and even packet radio subnets. It should also be obvious that the TCP/IP protocols will not, in all likelihood, interface with proprietary network protocols such as DECnet. For these systems/subnets to interface, with Internet protocols, there would need to be a TCP/IP implementation running on top of the native subnetwork protocol(s). There are currently several vendors which support the TCP/IP protocols on their hardware.

Robustness

The DARPA protocols were designed and have evolved to be robust in the face of node and subnetwork failure. This is a reflection on the military support of these Department of Defense protocols. It was envisioned that various portions of the Internet would be destroyed and the need to move data would remain. The Internet protocols adapt to sudden failures and attempt to deliver data to the end system, assuming it is still attached to the Internet. This is due directly to the fact that datagrams can and do take potentially different paths in reaching the end destination system. The network service is viewed as

inherently error prone and thus the transport layer protocol takes the responsibility to move data, end-to-end, reliably. A distinct advantage of this approach is that the much more complicated code to accomplish this function need only reside at the end systems, not at the intermediate ones.

High Performance

To some extent, the performance seen with the DARPA protocols will depend directly on the number of gateways traversed between end systems, and the bandwidth of those subnets that are traversed. In the face of congestion, a gateway running the TCP/IP protocols may drop packets, due to a lack of buffer space. This action may force retransmission of the dropped data packets. This loss of packets and subsequent retransmission is indigenous to connectionless protocols, where resources are dynamically allocated. The DARPA protocols also offer the user the ability to emphasize the type of delivery service required. The user can specify either an accurate or a quick delivery system. The accurate delivery system could consist of the TCP over the IP. In this configuration, the TCP would insure the data transmitted was received by the end system. The quick delivery system would involve using the UDP over IP; this is an unconfirmed delivery service.

Data Integrity and Security

The DARPA protocols do not directly support a data security package. That is, there are no encryption schemes built into the protocols. Data integrity is insured throught the use of the TCP over the IP. The IP by itself does not guarantee reliable transfer of data. Since the DARPA protocols were developed for the Department of Defense, data integrity was high on the list of requirements.

Transparency

Adding new nodes or subnets to the DARPA Internet has little impact on the already

established nodes and subnets. There is no need to take the Internet offline, and data transfer occurs as usual. If a new subnet is added, then it is possible that a routing table entry will be necessary for at least those core gateways which compose the main Internet. In fact, given the protocol used by the core gateways for exchanging routing information, it is likely only one of the core gateways would exchange this new routing information with its neighbor gateways. The addition of a new node or subnet can cause problems if the newly added routing information is incorrect. This sort of behavior has happened in the past, and can cause congestion by re-routing packets through the wrong gateways.

The DARPA protocols also handle transparency with regard to the fragmentation and reassembly of data while traversing the Internet. The basic method is to fragment data where necessary, to get data across a given subnetwork. Later, when the data arrives at the destination end system, the TCP module there will reassemble the data into its original form.

If a node, subnet, or gateway crashes or in some way fails, then the Internet handles this situation dynamically. through its protocols. Alternate routes will be found and the TCP will insure that all the data arrives at its destination.

Administrability

The Internet itself has the NIC, which administers the allocation of IP addresses to the main pieces of itself. It actually only allocates the network portion of the IP address, and lets the individual organizations allocate the host part of the address. If a group or groups wish to organize their own internet, then they are free to assign IP addresses as they see fit. However, Internet guidelines must be followed if the private internet is to ever attach to the Internet, due to the need for IP address uniqueness. There is not much support in the area of administrative tools for the Internet, and this has been dicussed by various researchers [13].

Traffic Accountability

The DARPA Internet protocols have been described as lacking in the area of traffic accountability. There are several reasons for this. First, the ability to bill end users was not high on the design requirements list for the Department of Defense. As a result of this, and due to the nature of the connectionless network service, traffic accountability is not easily supported. As was previously discussed, each gateway handles datagrams as separate entities, and that datagrams from the original block of user data may in fact traverse different paths enroute to the destination machine.

Chapter 6 - An Internetworking Problem and Proposal

A goal of this thesis is to provide a better solution to the problem of internetworking. Thus far, two important contributions to internetworking have been analyzed. Namely, the OSI and DARPA internetworking schemes. Rather than blindly suggest an arbitrary solution to an arbitrary internetworking problem, this thesis will identify a problem for which a solution may have real value, now and in the future. Therefore, before a problem is identified, an examination will be made of what the future of internetworking may hold.

The Future of Internetworking

It will be the purpose of this section to build a case for the long-term desirability and employment of the OSI protocols. First, an examination must be made to determine why the OSI protocols may well prevail over others, including the Internet protocols.

Users are mainly intereseted in services, not protocols [62]. It is no wonder then, why there would be great user interest in the OSI protocols. The newer, more recently developed OSI applications are significantly more functional (in theory) than those currently present in the Internet protocols. For example, the OSI file transfer and management protocol (FTAM) and the Internet file transfer protocol (FTP). Both allow simple transfer of files to and from heterogeneous systems, but in addition FTAM supports structured record files, and allows access to files with a greater granularity than is possible with FTP. The United States government has shown strong support for the OSI protocols. It has stated that at the beginning of the 1990s, requirements will be put in place to mandate the compliance of government purchased systems to the OSI protocols. As a result, vendors are already voicing their support for the OSI protocols in their products. Users are in support of the OSI protocols, in as much as they offer greater potential functionality, and the ability to operate over a wide variety of heterogeneous platforms. The ability to access data on different systems is highly valued by many users,

and is more an idea than a reality to the average computer user. The problem with the OSI protocols instantly taking over is the large embedded base of TCP/IP systems, which includes the Internet. As would be expected, people are viewing the OSI protocols as a threat to the money which has been invested in non-OSI compliant products. Also, there is the unknown; until OSI systems begin gaining popularity, users will be slow to invite the changes caused by moving to new software, and new protocols. The reality of this situation is that current users of the TCP/IP protocols will probably want and need to coexist with the OSI protocols. The period of transition will likely be a lengthy one. This is not an unusual condition. Old systems are often run in parallel with new systems until the bugs are worked out and the new system can be depended on.

From this point on, the assumption will be made that the OSI protocols will eventually dominate. The next section will therefore focus on a specific problem in the OSI internetworking scheme.

Problems with OSI

As was stated earlier, the OSI transport service supports five classes of transport protocols. A given OSI implementation need only employ one of the five protocols. Stated simply, the problem is this: if transport entity A requires the use of TP4, but transport entity B only supports TP0, then internetworking between the two will not be possible. Down negotiating among two transport entities is possible and allowable, but in the previous case TP4 is required, probably due to its local connection to a class C subnet. This is a major problem with the basic notion of OSI compliance and the implied internetworking which should accompany it. It would probably come as a shock to the average computer user that two protocol stacks, both of which support the OSI protocols, would be unable to internetwork. The next section will describe why this internetworking problem exists.

Transport and Network Layer Compatibility

The transport protocol used in a particular OSI implementation will depend largely on the underlying subnetwork. The transport classes 0 through 3 will only work properly with the CONS. These protocols depend on the fact that the underlying subnet is of class A or B, i.e. that they are *reliable* subnets. As such, classes 0 through 3 have no sophisticated procedures for error detection and recovery. Transport class 4 is the only transport protocol which will work with the CLNS. It assumes that the subnetwork is inherently error prone, and takes the necessary measures to make the transfer of data reliable. The internetworking problem now breaks down to the following:

- an OSI implementation need only employ 1 transport protocol
- it will only use as complex and costly a transport protocol as necessary
- it is not acceptable to run TP 4 on a CONS-based subnet, to facilitate internetworking ability

It is current practice that a transport user will employ the transport protocol barely necessary to accomplish data transfer, given the underlying subnetwork. Put another way, users of CONS-based subnets are not going to employ TP4, just to buy the ability to internetwork occassionally with an end system using TP4. This is due to several factors. First, TP4 is much more complicated a protocol to implement. Second, it is redundant when used above a CONS-based subnet. There is no need to employ an error detection and recovery protocol above a subnet which is reliable and handles the error detection and recovery for the transport layer. Third, this combination uses checksumming at both the subnet layer and the transport layer. Checksumming is generally known as a time and cpu consuming function. In short, running TP4 over a class A or B subnet is both time and resource consuming. The next section will suggest a three part solution to this problem.

Solutions for the Transport Layer Internetworking Problem

This section will offer a three part solution to the previously described internetworking problem. All three parts have their place in time:

- Use of a functional profile for universal internetworking
  [short-term solution]
- Use of transport bridges
  [medium-term solution]
- Mandate use of one transport protocol TP4
  [long-term solution]

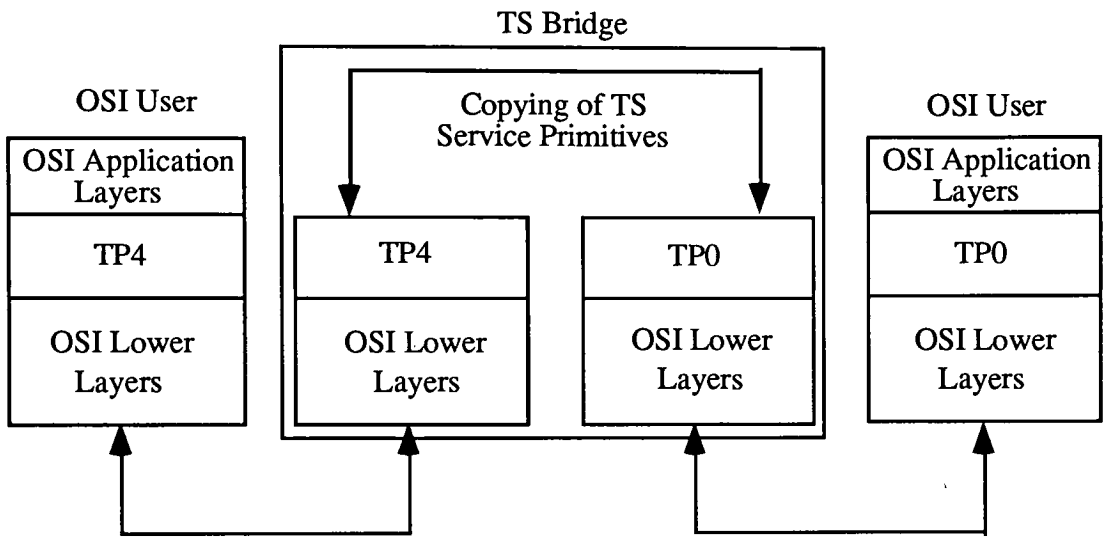
Use of a Functional Profile - SHORT-TERM SOLUTION

This solution is dubbed as the short-term solution as it requires no additional software or changes to the OSI standards. This solution to the internetworking problem takes note of the fact that several protocols exist for practically every layer of the OSI Reference Model. Therefore, a well-defined path is defined through the model. This path forms what is known as a functional profile. A good example of this solution can be found in the U.S. Government Open Systems Interconnection Profile (GOSIP) [69]. The GOSIP defines the protocols to be implemented at each layer, and may define how it is to be implemented. In this way, vendors and users have, in effect, a standard by which they can implement and interconnect OSI systems. The United Kingdom has also defined its own OSI profile. The drawback to this solution is that while it is useful for a single government or corporation, it does nothing to ensure that strict OSI conformace will guarantee internetworking capability.

Use of Transport Bridges - MEDIUM-TERM SOLUTION

The second solution to this problem is the one which is recommended, in the medium-term, by this thesis: the use of transport bridges. It is considered to be the medium-term solution as it requires the addition of functionality. This solution involves the idea, and use, of a transport service (TS) bridge. This is a relatively recent idea, and the history of

it is described in [62]. The purpose of the TS bridge is to copy service primitives from one class of transport protocol to another. Note that this scheme works because the service primitives do not vary among the various classes of the transport protocol. Therefore, the TS bridge acts as an intermediate store and forward node. The following diagram illustrates a TS bridge:



A Transport Service (TS) Bridge

TS Bridge Operation

TS Bridge operation is relatively straightforward. Application entity, A, wishes to communicate with an Application entity B. In this particular situation, internetworking is not possible, because the transport layer protocol associated with the Application entities is different, and cannot be down-negotiated. Thus, there is a need for a transport bridge function. This example will assume that the necessary bridging is as shown in the previous diagram. At the originating transport layer, a TSAP is setup. This TSAP is that of the end system where the target Application entity resides. The difference is that the originating transport layer will have the NSAP for the transport bridge system embedded in it. When the TS bridge receives the connection request, it will look up the target

TSAP, to see if it is valid for the bridge. If it is, then it will carry on a transport connection with the originating transport layer, and store the data as it is transferred. At the same time, the TS bridge will initiate a TC with the end system where the other Application entity resides. As data is transferred from the originating end system and passed through the bridge, it is immediately forwarded to the destination end system. Thus the data is transferred between the two end systems, through the TS bridge, in a manner not unlike the concatenation of two TCs. It should be noted that the TS bridge will operate transparently to the end system users. The one exception to this transparency may be the extra delay introduced by the bridge itself.

Problems with the TS Bridge

While it would appear that the TS bridge offers the ability to hurdle the internetworking problems associated with using different OSI transport protocols, there are several *new* problems which result:

- the TS bridge violates the rules of relaying
- it introduces another potential point of failure
- it impacts performance in a negative way
- there is potential for loss of data security
- there is potential for data corruption

The Rules of OSI Relays

The OSI reference model prohibits relays from existing above the network layer [35]. As such, the TS bridge is not an OSI supported entity. Clearly then, the OSI standards would need to be amended if this method is to be officially adopted. There are many good reasons for relaying being kept to the lower layers, and these reasons are discussed in the following paragraphs.

Introduction of a Failure Point

Since the TS bridge is of a store and forward nature, it inherently introduces another point of failure. Even if the two TS end users are functioning properly, if the TS bridge fails then the connection is lost. Basically, the TS bridge is just one more point at which failure can occur when two users are attempting to internetwork.

The Performance Impact

The introduction of a TS bridge in between two TS users negatively impacts the performance between them. The main reason for this is checksumming. Checksumming is already occuring at the two end TS entities, and the introduction of of an intermediate TS bridge means checksumming will also occur there as well. The act of computing a checksum is both time and resource consuming [62].

Security Loss

The TS bridge will also undo any attempts at encrypting data at the transport level. Since the TS bridge unpacks the data from it transport layer container, the actual data being passed becomes visible. If that data was previously encrypted by the originating TS entity, it becomes decrypted by the bridge before being re-encrypted and sent to the end TS entity. If data encryption occurs at levels above the transport layer, then this is a non-issue.

Data Corruption

Since the TS bridge sees the data in its clear form, it is possible that through a system malfunction it might corrupt the data as it passes through.

Mandated Use of TP4 - LONG-TERM SOLUTION

This solution is recommended as a long-term solution to the OSI transport protocol incompatibility problem. The reason it is a long term solution is that while this is technically feasible, the are technical considerations, and of course, politics.

This solution is somewhat reminiscent of the Internet method of internetworking: ONE TRANSPORT PROTOCOL ONLY. In this scheme, there is a single connection-mode transport layer protocol, the TCP. Internetworking is not a problem with the Internet protocols because of this. The OSI transport layer, on the other hand, supports five flavors of transport layer protocols. It seems that the ability to choose among many protocols has the negative side effect of making internetworking at best less than guranteed, and at worst difficult to achieve. Therefore, one solution to the problem would be to mandate the use of TP4 in order to be fully OSI compliant. This would seem to imply that the other classes of transport protocols, TP0 through TP3, would be dropped from the OSI transport standard. This is not the case, since the OSI transport protocols allow for negotiation of protocol class. Connections not requiring the higher overhead of TP4 could opt for a lower class protocol. Previously, the problems of using TP4 over class A or B subnets was described. It is very unlikely that this proposal would be accepted in the real world [62], as it stated that politically, administrators of CONS-based subnets are not interested in running TP4. It is hoped that in the long-term, hardware advances will be such that the higher overheads associated with TP4 will be a non-issue. This, along with the availability of tested TP4 implementations will hopefully make the univeral acceptance of TP4 a reality. The final hurdle to this solution is for the OSI committees to accept it and amend the OSI transport protocol standard.

Closing Remarks and Conclusion

This thesis purposely did not attempt to solve all the problems with internetworking OSI based systems. There are texts available [62] which outline in a much broader sense, the problems with the OSI protocols. As there were problems with the availability of too many protocol choices at the transport level, there too exists this problem at the network level. There, an implementor has the choice of either the CONS or the CLNS. And not

surprisingly, these two protocols will not allow internetworking, unless a relaying function between the two is implemented. However, the suggestion of a relaying function at the network layer is not going to break any OSI rules.

Conclusion

None of the solutions suggested can be considered to be both quick and final. The OSI standards were not developed overnight, and changing them will not occur overnight either. In the short-term, the use of functional profiles, while limiting total and universal internetworking, will work with a particular subset of those OSI systems which conform to them. In addition, no OSI standards support is required. Although the TS bridge would seem to bring along a whole new set of problems, it does solve a major problem: OSI compliant systems being unable to internetwork. There are incompatibility problems with the OSI transport layer protocols. One other advantage to the TS bridge solution is that of transition. The transition to OSI from other protocols was implied at the beginning of this chapter, where it was stated that ultimately the OSI protocols would be dominant. Because of the dollar investment alone, users will be slow to move toward the OSI protocols, especially if their system investments are at stake. Since there are so many TCP/IP-based systems currently, and it is believed there will be many more before OSI dominates, a transition plan from the Internet protocols to OSI is important. The TS bridge fits nicely into this transition. RFC 1006 describes a method for implementing TP0 above the TCP. In this way, it treats the TCP as a CONS. As of the writing of this thesis, RFC 1006 has already been implemented in LAN environments. This sort of a transition plan has another advantage: it allows engineers to implement and test OSI protocols above the transport level on the world's biggest internet! This is an extremely important fact. The Internet protocols did not attain the stability they now enjoy through use in small, controlled enviroments. They evolved through real use in a large interconnected environment, and their performance proves this out. To further this effort,

a package known as the ISO Development Environment as been developed by Marshall T. Rose [62]. This package consists of the upper four layers of the OSI protocols and is designed to interface to the TCP/IP. Finally, the idea of mandating the use of TP4 in OSI systems, while the most "final" of all the solutions, has the potential for the most conflict in reaching the point of an addendum to the transport protocol standard.

Chapter 7 - Epilogue

This thesis purposely selected a specific internetworking issue, to illustrate and for which to propose a solution. This was in keeping with one of the primary goals of the thesis. In the course of researching, several other internetworking issues were found, and to this point have received only brief mention. It will be the purpose of this section to list and describe these issues in slightly more detail. As was the case with the previous chapter, the assumption that the ISO protocols will eventually prevail will be made. The following areas present potential problems with regard to internetworking with the ISO protocols:

- Lack of Routing Protocols
- Lack of Address Resolution Protocols
- Two Different Network Protocols

Lack of Routing Protocols

Routing protocols are inportant to any internetworking schemes. These protocols provide a means for the collection and distribution of routing information. To date, no ISO standards have been issued for this purpose. This presents a problem for implementors of OSI protocol based systems. Compare this to the situation with the DARPA Internet protocols. There are several Internet protocols which are used to implement the collection and distribution of routing information. They are the Gateway-to-Gateway Protocol (GGP), the Exterior Gateway Protocol (EGP), and the Interior Gateway Protocol (IGP). These have been described in previous chapters. It can be seen from the descriptions that there is a great deal of cooperation required for the routing protocols to work effectively. This enforces the notion that protocols such as these be agreed upon, implemented, and used. One can only imagine the various, possibly differing schemes which OSI implementors may use to accomplish this. Even worse, they may just default to using static routing tables and avoid the implementation of routing protocols

altogether. This would likely have a negative impact on the robustness of a given implementation. For example, consider what would happen with static routing tables if an intermediate system crashes and is inoperative for a period of time. Even if the another intermediate system is used (as part of a dynamic mechanism), how does the end system know when the crashed machine becomes available again? At any rate, the lack of ISO standards in this area may result in a wide variety routing protocols implementations (a bad thing), or the use of a single, de facto standard (a better thing). There may also be problems with adopting some of the Internet protocols as ISO standards. Consider the EGP, for example. It currently requires that all the network reachability information it passes fit into a single IP datagram. As an internetwork grows to a large size, this becomes a serious consideration. There is currently an effort underway to explore and hopefully remove this restriction [14]. A short-term measure, known as "subnetting", also offers a method for decreasing the ratio of subnets to IP network addresses. This has the effect of having to distribute potentially less routing information for a given number of attached subnets.

Another area for improvement in routing protocols concerns the metrics used to measure distance between an end system and the gateways between it and the destination end system. The Internet protocols use a metric known as "number of hops". This refers to the number of gateways which have to be traversed to a given destination end system. This metric has certain inherent disadvantages. For example, consider a route to a subnet which has a number of hops metric equal to 1. There is no guarantee that the hop to this gateway is a high speed link, or a 300 baud serial line. Yet if another path to the same subnet is listed as having a 3 hop metric, then the 1 hop route will be chosen over the 3 hop. What is required is a method of encoding other data such as link speed, delay, cost, or reliability. There is current research ongoing in the area of defining new metric values for use in routing protocols [14].

One last consideration for improving routing protocols is the ability to handle mobile hosts. Current Internet and ISO addressing schemes are such that addresses are assigned by the local domain administrator when the system is added to the internet. If the system is physically moved, say to another subnet, it will likely require a new IP address. The protocols used currently do not dynamically assign addresses, such that a host could move from site to site and enjoy uninterrupted use. Mobile hosts will demand a dynamic, real-time protocol scheme for routing updates and possibly address re-assignment. This is turn has a significant impact on the speed at which all gateways will need to keep their routing tables updated. A need for globally unique transport connection identification is discussed with regard to mobile hosts in [15]. The scheme discussed in that paper describes a single database which would contain dynamic routing information. This routing data would presumably be updated as the mobile host moved from subnet to subnet. It should be noted that this is currently an important need for the military community.

Lack of Address Resolution Protocols

As has been stated previously, there is a general lack of address resolution protocols in the OSI internetworking scheme [62]. Address resolution protocols are used to determine a physical subnetwork address from a logical network address. It is interesting to note that the OSI standards have attempted to keep routing and physical address information disjoint from the logical network address. That is, there are no methods outlined by standards for determining a physical address of an end or intermediate system from a logical address. In reality, the actual physical address of the system in question may be encoded in the domain specific part (DSP) of the NSAP (the logical address, in this case). The DARPA Internet has a protocol known as the Address Resolution Protocol (ARP). It functions as follows: when a gateway realizes the next gateway or destination end system

resides on the current subnet, it sends out an ARP request. All nodes on the subnet may see this request, but only the system in question responds to it. The response contains the physical address of the end system it wishes to forward data to. Some of the general issues to consider in implementing an ARP are:

- how many times should a system be requested to supply the necessary address information?
- should responses to previous ARP requests be saved (cached), and if so, for how long?

A disadvantage resulting from the lack of address resolution protocols in the OSI scheme is the amount of routing data required for a given system. The Internet has already seen the need for limiting the amount of routing information required per system, as the routing information must be distributed to a large number of systems.

Two Different Network Protocols

There exists a potential internetworking problem at the network layer in the OSI scheme. It is a problem, similar in some ways, to the one previously described with regard to the ISO Transport protocols. This problem results from the specification of two different protocols at the network layer. The two protocols are the CONS and the CLNS. To illustrate the problem, consider two fully OSI compliant implementations of the ISO protocols. Both implementations are composed of identical protocols, except that one implementation uses the CONS, while the other uses the CLNS. Two systems specified like this will not internetwork! However, unlike the situation at the transport layer, relays are allowed to exist at the network layer. In this way, an intermediate system may be used to form a union between two end systems which use the two different network protocols. ISO 8648 [35] describes the architecture scenarios that may be used to implemement the CONS and the CLNS over a combination of CONS and CLNS based subnetworks. In closing, the fact must be stated that here is another example of an OSI

internetworking problem introduced by having too many choices. Although this problem can be overcome, it is a problem which simply does not exist in the Internet world. This fact can be directly attributed to the fact that the there are fewer choices within the Internet with regard to network and transport protocols.

Chapter 8 - Literature Search

In order to compile a list of articles on which to base this thesis, a literature search was conducted. This search was executed on the premises of Eastman Kodak, at the Apparatus Division Library in Rochester, NY. Rich Bartl, the librarian there, met with me to discuss my research needs and explained the resources available. The principle literature search took place via three bibliographic indices. They were NTIS (National Technical Information Service), Compendex (engineering index and engineering meetings), and Inspec (by Institute of Electrical Engineers, London). These databases encompass many of the computer science and electrical engineering journals. The search went back to journals printed on or after Jan 1, 1970. The search criteria was for a correlation between:

- internets
- internet
- internetting
- internetwork
- communications

plus other forms of the words or index terms.

Approximately 100 articles matched the keywords used. From these 100, 38 articles were obtained after reading the accompanying abstract for each. Additional articles/texts were obtained from the bibliographies contained in the above articles. The ISO standards used for background in this thesis were obtained from the Standards Library at the Kodak Research Lab.

Chapter 9 - Glossary

ARPA

The Advanced Research Projects Agency. This was the precursor to DARPA. This is the group that initiated the TCP/IP effort.

bridge

A type of relay which functions to store and forward data frames between LANs. In a general sense, a relay at any level of the Reference Model which serves only to store and forward information, without special knowledge of the underlying layer.

congestion

An undesirable feature of a given internetworking design. This occurs when a destination or subdestination node cannot handle the network traffic, causing data packets to back up.

CLNS

The ConnectionLess Network Service. This is the OSI network service in which unrelated data units are passed along by network layer entities. In this scheme, no connection establishment is required to send a data unit to another network layer entity.

CONS

The Connection Oriented Network Service. This is the OSI network service in which virtual circuits are set up between network layer entities. In this scheme, a coonection must first be arranged, before data may be transferred,

CCITT

The Telegraph and Telephone Consultative Committee.

DARPA

The Defense Advanced Research Projects Agency.

DDN

Defense Data Network. That portion of the DARPA Internet which is administered by the Department of Defense. This includes the ARPANET as well as several military nets.

flow control

The act of controlling the data packet traffic between two points in an internetwork so that congestion is minimized.

gateway

A type of relay which functions to store and forward data packets between LANS.

IP

Internet Protocol. One of the DDN Protocols.

Internet

The capitalized form of this word is commonly used to refer to the DARPA Internet.

internet

The interconnection of two or more networks for the purpose(s) of resource sharing, communications, and/or to provide a single administration point.

ISO

International Organization for Standardization.

LAN

Local Area Network.

NS

Network Service. This term is used in the OSI standards documents.

NSDU

A Network Service Data Unit. This is the protocol data unit exchanged between two OSI network service providers.

octet

A unit of data having a length of 8 bits. This term is most often used when discussing DARPA Internet datagrams.

OSI

Open Systems Interconnection. The goal of this effort to define standards for data communication which will allow computers from different vendors to communicate with each other. One visible result of this effort in the OSI Reference Model.

OSI Reference Model

This is the seven layer architecture for communication between computer systems. It is the basis for the OSI service and protocol definitions.

PDU

Protocol Data Unit. In a general sense, this term may be applied to the protocol data units (information) passed between any two peers within the Reference Model.

protocol converter

A type of relay which functions to convert from one protocol to another across two networks.

QOS

Quality of Service. This term is comonly applied to a set of parameters which may be passed from Transport service user, down to the network level. It is used to convey the quality of communication deemed necessary.

relay

A generic term used to describe an entity which is used to convert data packets as they pass from one network to another.

repeater

A type of relay which functions to copy bits between cable segments of network.

It can be used to increase the effective length of a network cable.

TCP

Transmission Control Protocol. One of the Internet Protocols.

UT

Universal Time. This is generally accepted as the standard time in all international affairs. It is defined as the time at Greenwich, England, as expressed in a 24-hour format.

X.25

Layers 1, 2, and 3 of the CCITT international standard network access protocols.

Chapter 10 - Bibliography

[1] Bauman, T. M., "Subnet access technologies and internetworking within the TMN", *Proceedings of the IEEE INFOCOM '88 7th Annual Joint Conference*, pp. 429-432, March 1988.

[2] Benhamou, E., and Estrin, J. "Multilevel Internetworking Gateways: Architecture and Applications", *IEEE Computer*, vol. 16, no. 9, pp. 27-34, Sept.1983.

[3] Bird, D., "A question of communication", *Systems International Networking Extra*", pp. 5-10, Feb. 1988.

[4] Bird, D., "Internetworking", *Database and Network Journal*, vol. 18,no. 1, pp. 2-8, 1988.

[5] Boggs, D. R., Shoch, J. F., and Taft, E. A., "PUP: An Internetwork Architecture", *IEEE Transactions on Communications,* vol. COM-28, no. 4, pp. 612-624, April 1980.

This paper is a good, early reference on internetworkling in general, Pup in particular.

[6] Bozetti, M., and Ravasio,P. C., "Internetworking Among Local and Long Haul Networks: A Case Study", *Proceedings of the 5th Conference on Computer Communication*, pp. 729-734, Oct. 1980.

[7] Burg, F. M. and Chen, C. T., "Of local networks, protocols, and the OSI reference model", *Data Communications*, pp. 129-150, Nov.1984.

[8] Cerf, V.,and Kahn, R., "A Protocol for Packet Network Intercommunication", *IEEE Transactions on Communications,* Vol. COM-22, No. 5, May 1974, pp. 637-648.

An early paper on packet networking; Cerf is an early pioneer of Internet protocols. Mandatory reading for the inter/networking student.

[9] Chadwick, H. D., and Gulick,A. M., "High-speed internetworking in a multi-vendor environment", *IEEE Military Communications* Conference, vol.1, pp. 181-186, 1988.

[10] Chen, W.-T., Huang, N.-F., and Chen, Y.-Y., "The Design of an Internetwork", *Proceedings of the Pacific Communications Computer Symposium,* pp. 505-509, Oct. 1985.

[11] Chiou, I.Y., and Liu, M.T., "CAMPUSNET: A Gateway-Network Approach to Interconnecting a Campus-wide Internet", *Proceedings of the IEEE INFOCOM*, pp. 168-177, 1985.

[12] Chiou, I.Y., and Liu, M.T., "GATENET: A Voice/Data Internet Transport System", *IEEE INFOCOM '86, 5th Annual Conference "Computers and Communications Integration Design, Analysis, Management*", pp. 39-46, April 1986.

[13] Clark, D.,"The design philosophy of the DARPA Internet protocols", *Computer Communications Review*, vol. 18, no. 4, pp. 106-114, Aug. 1988.

An excellent paper which discusses the history of the DARPA Internet. It

describes the design goals, successes, and failures. This is mandatory reading for the student of internetworking.

[14] Comer, D.E., *Internetworking with TCP/IP: principles, protocols, and architecture*, New Jersey: Prentice-Hall, 1988.

This is an excellent source of Internet protocol information and description. It also includes a history of the Internet, lists all the RFCs (by category, by number), explains how to obtain RFCs, and discusses some of the current research going on within (and wihtout) the Internet community. This is mandatory reading for those who find themselves reading the RFCs concerning the main Internet protocols.

[15] Davies, B. H., and Bates,A. S., *Internetworking in the Military Environment*, Malvern (England), 32 pp. July 1981.

[16] Day, J. D. and Zimmermann, H., "The OSI Reference Model", *Proceedings of the IEEE*, vol. 71, no. 12, Dec 1983.

This paper contains a breif description of the OSI reference model. A good paper for those uninitiated into the OSI model, wanting an explanation without a lot of OSI technical jargon.

[17] De, S., and Xing-gang, W., "Internetworking and Protocol Conversion", *Proceedings of the 7th International Conference on Computer Communication*, pp. 345-351, Oct.1985.

[18] Elam, G. B., Liu, Y. J., and Raimer, K.A., "LAN/WAN internetworking performance issues", *IEEE Military Communications Conference*, vol. 1, pp. 187-192, 1988.

[19] Foley, J. S., "The status and direction of open systems interconnection", *Data Communications*, pp. 177-193, Feb. 1985.

[20] Folts, H.C., "Coming of age: A long-awaited standard for heterogeneous nets", *Data Communications*, pp. 63-73, Jan. 1981.

[21] Garg, A. R., "LAN Internets : Addressing the Issues", *MIDCON*, pp. 1-6, 1982.

[22] Groenbaek,I., "Conversion Between the TCP and ISO Transport Protocols as a Method of Achieving Interoperability Between Data Communications Systems", *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, no. 2, Mar. 1986.

This paper outlines a method for building a TCP <-> ISO protocol converter. Interesting reading, especially of those concerned with internetworking problems at the transport level.

[23] Hedrick,C. L., "Introduction to the Internet Protocols", *Computer Science Facilities Group*, Rutgers University, July 1987.

This paper provides a general background on the Internet protocols; especially good for the uninitiated reader.

[24] Henshall,J., and Shaw, S., *OSI Explained: End-to-End Computer Communication Standards*, England:Ellis Horwood Limited, 1988.

Texts on OSI are currently sparse; this text provided helpful background information. It is mainly concerned with the OSI protocols which reside above the transport layer.

[25] ISO 7498, "Open Systems Interconnection - Basic Reference Model".

Mandatory reading for the OSI internetworking student.

[26] ISO 8072, "Transport Service Definition".

[27] ISO 8072/AD1, "Transport Service Definition, Addendum 1."

This addendum includes definition of a connectionless transport service.

[28] ISO 8073, "Connection Oriented Transport Protocol Specification".

[29] ISO 8073/AD1, "Connection Oriented Transport Protocol Specification, Addendum 1 "

This addendum includes the connectionless mode transport service definition.

[30] ISO 8073/AD2, "Connection Oriented Transport Protocol Specification, Addendum 2".

This addendum includes class four operation over a connectionless network service.

[31] ISO 8208, "X.25 Packet Level Protocol For Data Terminal Equipment".

Mandatory reading for the OSI internetworking student.

[32] ISO 8348, "Network Service Definition".

Mandatory reading for the OSI internetworking student.

[33] ISO 8473, "Protocol For Providing the Connectionless-mode Network Service".

Mandatory reading for the OSI internetworking student.

[34] ISO 8602, "Protocol For Providing the Connectionless-mode Transport Service".

Mandatory reading for the OSI internetworking student.

[35] ISO 8648, "Internal Organization of the Network Layer".

Mandatory reading for the OSI internetworking student.

[36] ISO 8878, "Use of X.25 to Provide the OSI Connection Mode Network Service".

Mandatory reading for the OSI internetworking student.

[37] ISO 8880, "Specification of Protocols to Provide and Support the OSI Network Service".

Mandatory reading for the OSI internetworking student.

[38] ISO 8881, "Use of the X.25 Packet Level Protocol in Local Area Networks".

[39] Klerer, S. M., "The OSI Management Architecture: an Overview", *IEEE Network*, vol. 2, no. 2, pp. 20-29, Mar. 1988.

[40] Knightson, K. G., Knowles, T., and Larmouth, J., *Standards for Open Systems Interconnection*, U.S.A.: McGraw-Hill, 1987.

Texts on OSI are currently sparse; this text provided background information on the OSI model, services, and protocols. I did not, however, like the type of paper it was printed on.

[41] Lea, C. -T., "A high performance LAN internetwork design", *IEEE Global Telecommunications Conference*, vol. 1, pp. 19-23, 1986.

[42] Longman, T. C., "Open architecture internetworking", *Proceedings of the LOCALNET Conference*, pp. 55-67, Oct. 1985.

[43] Martinez, R., "Internet gateway design for Defense Data Network access", *IEEE Military Communications Conference*, vol. 1, pp. 15.4.1-5, 1986.

[44] Perry, D.G., Blumenthal,S. H.,and Hinden, R. M., "The ARPANET and the DARPA Internet", *Library HiTech*, vol. 6. no. 2, pp. 51-62, 1988.

Another excellent paper on Internet history and evolution. Mandatory reading.

[45] Piscetello, D. M., Weissberger, A. J., Stein, S. A. and Chapin, A. L., "Internetworking in an OSI environment", *Data Communications*, pp.118-136, May 1986.

[46] Postel, J. B., "Internetwork protocol approaches", *IEEE Transactions on Communications*, Vol. COM-28, no. 4, pp. 604-611, April 1980.

A concise paper which describes virtual circuit (X.75) and datagram (DARPA) internetworks. An early paper by a notable internetworking author.

[47] Quarterman, J. S.and Hoskins, J.C., "Notable Computer Networks", *Communications of the ACM*, vol. 29, no. 10, pp. 932-971, Oct.1986.

A rather lengthy paper which discusses a broad variety of inter/networks. Even the social implications of these communication media are discussed. An excellent background document.

[48] RFC 790, Postel, J.,"Assigned Numbers", USC/Information Sciences Institute, September 1981.

[49] RFC 791, "Internet Protocol", Sept. 1981.

This RFC describes the Intemet Protocol (IP) in full detail; mandatory reading for the internetworking student.

[50] RFC 792, Postel, J. B., "Internet Control Message Protocol", Sept. 1981.

This RFC explains the ICMP in full detail; it is mandatory reading for the student of internetworking.

[51] RFC 793, "Transmission Control Protocol", *Defense Data Network Protocol Handbook*, vol. 2, pp. 2.179-2.198, Sept. 1981.

This RFC fully describes the TCP; it is mandatory reading for the student of Internet protocols.

[52] RFC 823, "The DARPA Internet Gateway", Hinden, R., and Sheltzer,A., Sept. 1982.

[53] RFC 826, "An Ethernet Address Resolution Protocol", David C. Plummer, November 1982.

[54] RFC 888, "Stub Exterior Gateway Protocol", Seamonson, L. J., and Rosen,E. C., Jan. 1984.

This is the first RFC which describes the EGP.

[55] RFC 904, "Exterior Gateway Protocol Formal Specification", Mills, D. L., April 1984.

This RFC describes the EGP, which will be of interest to students of Internet gateways, and those requiring a knowledge of routing table dispeersion.

[56] RFC 975, "Autonomous Confederations", Mills, D. L., Feb. 1986.

[57] RFC 983, "ISO Transport Services on Top of the TCP", Cass, D.E. , and Rose, M. T., April 1986.

This RFC describes the first version of the ISO Transport services above the TCP. RFC 1006 describes the latest version

[58] RFC 985, "Requirement for Internet Gateways- Draft", May 1986. Prepared by the Gateway Requirements Subcommittee of the National Science Foundation's Network Technical Advisory Group, David L. Mills, chair.

Another good background paper for those interested in Internet gateways.

[59] RFC 1006, "ISO Transport Services on Top of the TCP", May 1987.

This RFC specifies version 3 of the protocol and supersedes RFC 983. This reference is interesting reading for individuals wishing to run OSI application layers above the TCP end to end protocol.

[60] RFC 1009, "Requirements for Internet Gateways", Braden, R., and Postel, J., June 1987.

An updated RFC for those interested in Internet gateway operation/implementation.

[61] Rose, M. T. and Cass, D. E., "OSI Transport Services on Top of the TCP", *Computer*

Networks and ISDN Systems, vol. 12, pp. 159-173, 1987.

This paper was useful, as it describes a method for running OSI applications over the TCP transport service. Rose is a brand-name author in the TCP/OSI internetworking arena.

[62] Rose, Marshall T., "The Open Book: A Perspective on OSI", New Jersey: Prentice-Hall, 1990.

This is an excellent text! It is current, up-to-date, and written in style that allows the technical issues to be understood, along with the political reasons for the designs. The author is very candid about his opinions, and clearly defines his own opinion from the facts. Everyone interested in the aspects of the OSI reference model, services, and protocols should read this text. Mandatory reading.

[63] Sablatash, M., "Problems and Possible Approaches to the Interworking of Large Computer Networks", *Proceedings of the 2nd International Symposium on Large Engineering Systems*, pp. 9-14, May 1978.

[64] Shimell, P.F., "Gateways, road blocks, access points, and a route guide to better internetworking", *Proceedings of Videotex '84 International*, pp. 283-298, Nov. 1984.

[65] Solomon, C., "Exploring the problems of internetworking", *Data Communications*, vol. 14, no. 7, pp. 177-189, June 1985.

This paper was useful for deriving the functional requirements for internetworks.

[66] Stallings, W., *Handbook of Computer-Communications Standards, Volume I*, Indiana: Howard W. Sams & Company, 1989.

Texts on OSI are currently sparse; this book covers the OSI layers in detail, and was a very useful guide, while reading the actual OSI standards documents.

[67] Sung, K., "Why do we have problems designing and implementing with 'military standard internet protocol (mil-std-1777)'?", *IEEE Fifth Annual International Phoenix Conference*, pp. 224-229, 1986.

[68] Tanenbaum, A. S., *Computer Networks*, New Jersey: Prentice-Hall, Second Edition, 1988.

This is a good, general text on the subject of computer networking.

[69] U.S. Government Open Systems Interconnection Profile (GOSIP). U.S. Federal Information Processing Standards Publication 146, April 1989.

This document describes the functional profile to be used by the US Government. As such, it is an example of the short-term solution proposed by this thesis.

[70] Weissberger, A. J. and Israel, J. E., "What the new internetworking standards provide", *Data Communications*, pp. 141-156, Feb.1987.

[71] X.25, "Interface between data terminal equipment (DTE) and data circuit terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuits", International Telephone and Telegraph Consultative

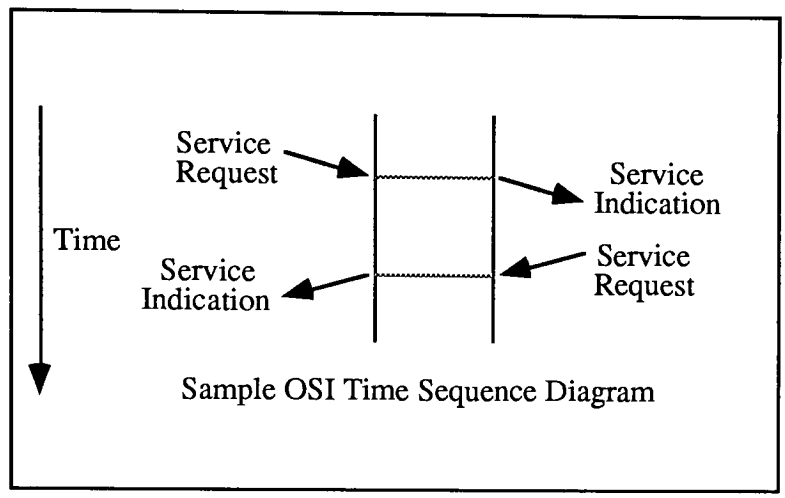
Committee (CCITT), 1988.

This standard was important reading as it describes a protocol which very nearly provides the OSI CONS, not to mention that in general, it is a commonly used protocol.

[72] Yu, A., Atwood, J. W., and Radhakrishnan, T., "Enhancing a local area network for internetworking", *IEEE 12th Conference on Local Computer Networks*, pp. 66-71, 1987.

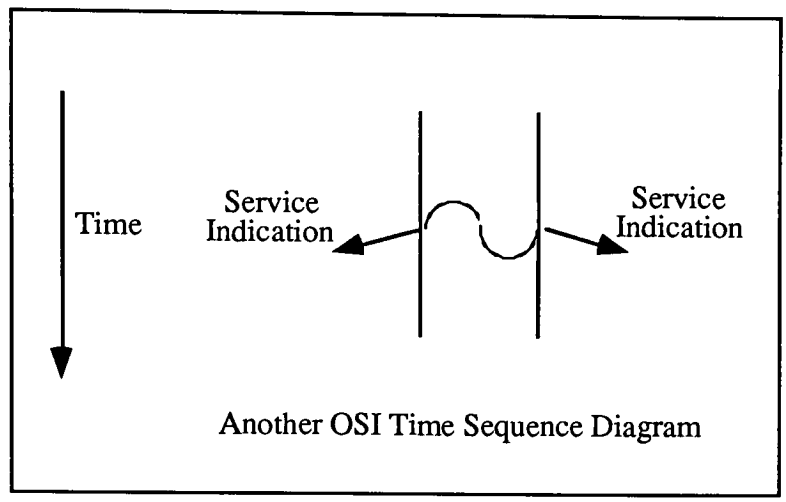
Appendix A

This appendix describes how to interpret typical OSI event sequence diagrams.



This diagram is meant to show the exchange of information between the two service users and the service providers. The two vertical lines show the boundary between the service user and service provider. Time increases in the downward direction. For example, in the above diagram, it can be stated that there is an exchange of information, via a service provider, between two service users. It can also be stated that the service request initiated the dialogue between the users, followed closely by the service indication, service response, and service confirmation.

Another common time sequence structure is demontrated by the following diagram:



Another OSI Time Sequence Diagram

In this diagram, no information is passed by the service users, to each other. Rather, information originates with the service provider, and is passed to the service users. The curly line between the vertical user/provider boundary denotes the fact that no data is passed across between users, and signifies a transfer of information between the service provider and service user only.