

TCP/IP Troubleshooting

Editorial Number 1



Extract from IDC Technologies' Practical TCP/IP and Ethernet Networking for Engineers and Technicians Workshop

© Copyright IDC Technologies, February 2000



SPECIALISED ENGINEERING, CONSULTING AND TRAINING SERVICES FOR INDUSTRY

Australia • Canada • Ireland • Malaysia • New Zealand • Singapore • South Africa • United States • United Kingdom

Contents

1.	Introduction to TCP/IP
1.1	Introduction to TCP/IP
1.2	What is TCP/IP?
1.3	Open Systems
1.4	ARPA Architecture
1.5	OSI Model
1.6	TCP/IP (or DoD) Model
2.	Common TCP/IP Protocols
2.1	Telecommunications Network (TELNET)
2.2	File Transfer Protocol (FTP)
2.3	Simple Mail Transfer Protocol (SMTP)
2.4	Kerberos
2.5	Domain Name Service (DNS)
2.6	Simple Network Management Protocol (SNMP)
2.7	Network File System (NFS)
2.8	Remote Procedure Call (RPC)
2.9	Trivial File Transfer Protocol (TFTP)
2.10	Internet Control Message Protocol (ICMP)
3.	Routing : IP
3.1	Routing Defined
3.2	Routing and the Physical Address
3.3	Field Destination
3.4	IPv6
4.	Network Interface Connection
4.1	Ethernet
4.2	Frame Relay Example
4.3	Troubleshooting the Network Interface
5.	Internet Connection
6.	Host to Host Connection
6.1	Transmission Control Protocol (TCP)
6.2	User Datagram Protocol
7.	Process and Application
7.1	SNMP Example
7.2	Troubleshooting the Process/Application Layer

Foreword

It is hoped that this write-up on TCP/IP will be of use to you. It is an extract from the preliminary manual for our workshop on Practical TCP/IP for Engineers and Technicians and gives a broad overview of the topic.

The write-up is broken up into the following sections:

Introduction to TCP/IP

This gives an overview of this very important family of protocols.

Common TCP/IP Protocols

Some of the more important protocols are discussed here.

Routing: IP

This is one of the more important strengths of the TCP/IP family of protocols and is reviewed.

Network Interface Connection

This refers to the physical layer and must be configured correctly to operate satisfactorily.

Internet Connection

One of the difficulties here is to ensure that the IP addresses are assigned correctly.

Host to Host Connection

The two protocols UDP and TCP used here are discussed.

Process and Application

This sits on the top of the protocol stack and can be accessed by FTP and TELNET, for example.

Should you have any queries on the material following; please do not hesitate to give us a call. We would be delighted to assist you where possible.

1. Introduction to TCP/IP

1.1 Introduction to TCP/IP

TCP/IP is the networking protocol of choice today. It is widely used in the UNIX and Intranet communities. Smaller PC-based systems are migrating to TCP/IP in larger numbers as their networking protocol of choice.

1.2 What is TCP/IP?

The term "TCP/IP" is an acronym made of the two fundamental protocols that are part of a larger set. These protocols constitute a set of standards that together offer a strong networking solution for a large variety of operating systems, from PC machines running DOS and Windows, to midrange mini computers running UNIX and mainframes running operating systems such as MVS and VMS among others. TCP stands for **Transmission Control Protocol**, while IP is the **Internet Protocol**. TCP/IP is closely aligned with the UNIX operating system, with much of the original work on the protocols done using UNIX systems. This work was primarily done by the people involved with the ARPANET and DARPA (see below) in the United States on behalf of the US Military and US government. TCP/IP is constantly evolving, with one of the more public recent changes being the need for an updated version of IP to allow more Internet addresses to be used.

TCP/IP evolved from work done in the late 1960s and early 1970s by the Advanced Research Projects Agency Network (ARPANET) and US Defence Advanced Research Projects Agency (DARPA) to test and check the viability of packet switching technology. There was also a requirement to develop a standardised communications procedure (open system) that would inevitably be used on a variety of platforms.

1.3 Open Systems

An example of an Open System is the OSI 7-layer reference model which is covered in detail in the IDC LAN course. A decade ago open systems were virtually non-existent. Each hardware manufacturer had a product line and you were bound to that supplier for all your hardware and software needs. Some companies took advantage of this situation and charged outrageous prices, or forced unwanted configurations on their customers. The groundswell of public opinion started to force companies into accepting open systems principles. UNIX is a classic example of an open software system platform. The source code for the UNIX operating system has been readily available to anyone who wanted it almost from the start.

1.4 ARPA Architecture

The ARPA Internet architecture consists of four layers. The lowest layer is the Network Interface Layer that comprises the physical link between devices. The next layer is the Internet layer that acts as a buffer between the host and the network-specifics of the underlying hardware.

Delivery is not guaranteed with IP and a Service Layer was added to provide a form of reliability. There are two protocols that fall into this category. TCP is the protocol used when extremely reliable delivery is necessary, while the User Datagram Protocol (UDP) is used when unreliable delivery is suitable (usually with the higher level layer or application providing whatever reliability is necessary). The highest TCP/IP layer is the one used for the host application and includes applications such as TELNET (Telecommunications Network), FTP (File Transfer Protocol) and SMTP (Simple Mail Transfer Protocol).

OSI Layer	ARPA Architecture
Application	Process / Application Layer
Presentation	
Session	
Transport	Host-to-Host Layer
Network	Internet Layer
Data Link	Network Interface or Local Network Layer
Physical	

Figure 1
Comparison of OSI and ARPA Models

The figure above shows these four TCP/IP (ARPA) layers compared to the **Open Systems Interconnection (OSI)** seven-layer reference model.

The ARPA model was designed to connect hosts serving the academic, research, government and military populations, primarily in the United States. The OSI Reference Model was broader in scope and was the product of an international standards organisation and included input from people in Europe and Asia as well as the United States. In summary the ARPA model was more specific while the OSI model was more general in nature.

1.5 OSI Model

The Open System Interconnection model is a suggested reference model which isolates various parts of the networking layers from all other layers other than the two layers directly connected to it. Below is an example of the OSI Reference Model, from which it can be easily seen that (for example) the Transport Layer only has to communicate with the Network and Session layers. All other layers are not directly linked to the Transport layer, so that the Transport layer has no direct impact on (and receives no direct communications from) other layers.

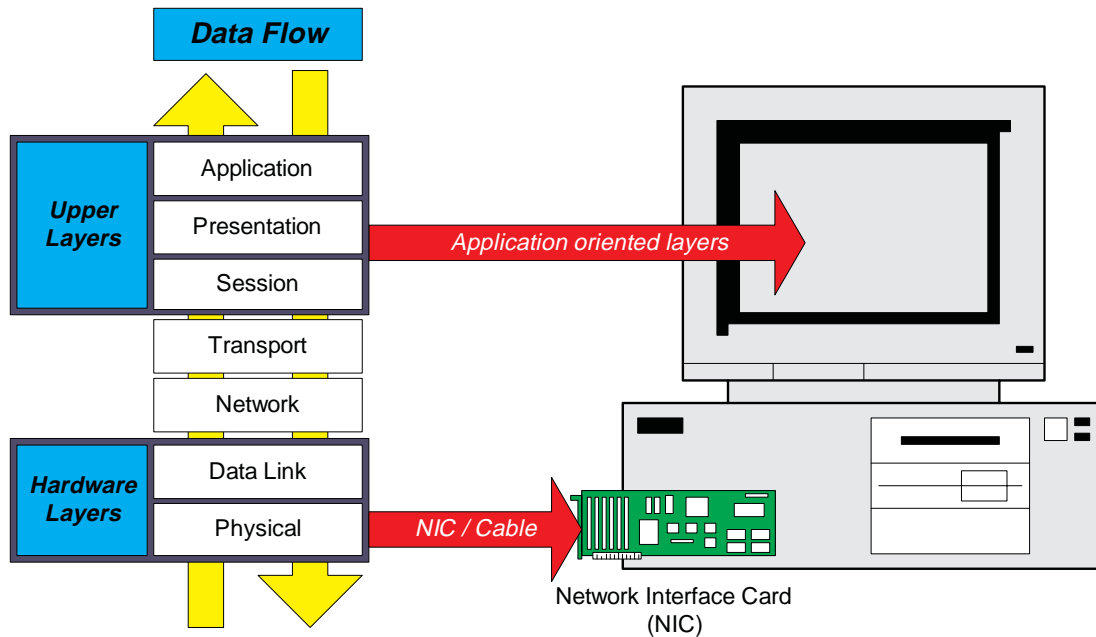


Figure 2
OSI Model Layers

1.6 TCP/IP (or DoD) Model

The TCP/IP model differs from the OSI model in that there are only four layers in the TCP/IP model. The term **DoD** is an acronym for “Department of Defence”, which where the original TCP/IP protocol suite was developed in the United States. This term is included here for completeness, although not many people use the DoD acronym now. TCP/IP and the OSI models were developed concurrently, and each model has contributed to the development of the other.

The basic requirements for TCP/IP were:

- A common set of applications
- Dynamic routing
- Connectionless protocols at the networking level
- Universal connectivity
- Packet switching

TCP/IP combines the (OSI) session and presentation layers into the TCP/IP application layer. In addition, because TCP/IP was required to provide a connectionless service, the (OSI) physical layer and data link layer were combined into the one TCP/IP network layer.

TCP/IP therefore has only four layers as illustrated below:

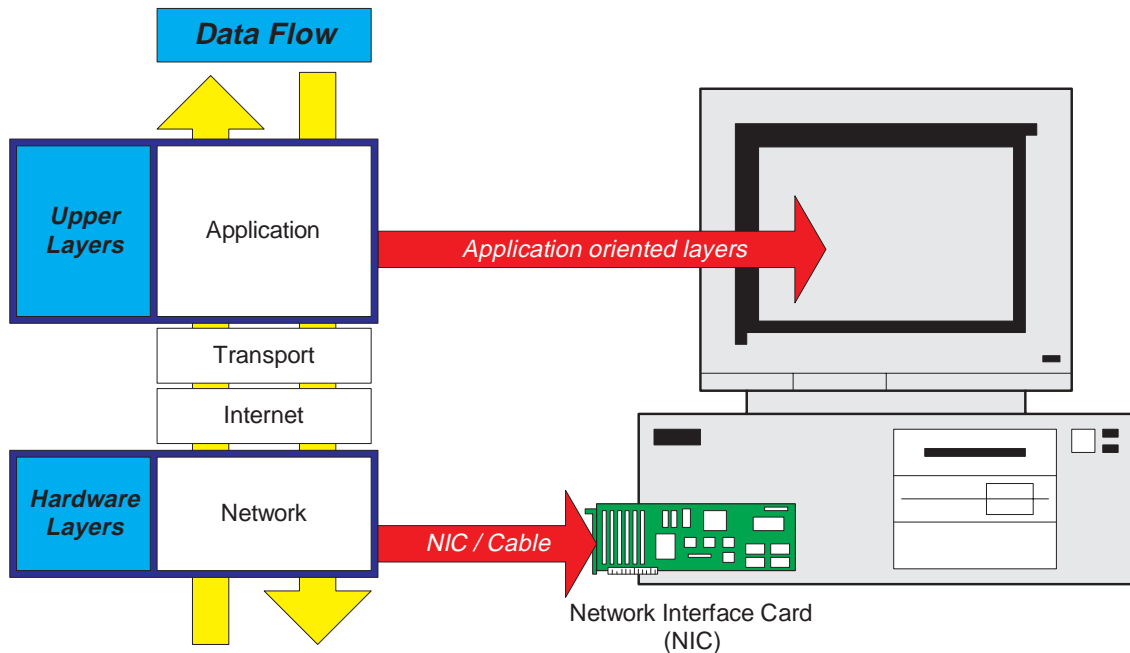


Figure 3
TCP/IP Layers

The four layers that make up the TCP/IP model are explained below:

Application Layer

At the highest level the application programs interact with the transport level protocol to send or receive data.

Transport Layer

The transport layer provides end to end communications. It ensures that data arrives without errors and in the correct sequence.

Internet Layer

This performs the encapsulation of the request from the transport layer in an IP datagram, uses the routing algorithms to determine whether to deliver the datagram directly or send it to a gateway. The internet layer sends ICMP error and control messages as needed.

Network Interface or Physical Layer

This is responsible for accepting IP datagrams and transmitting them over a specific network.

2. Common TCP/IP Protocols

A quick overview of some of the more common protocols that make up the TCP/IP protocol suite follows. Some of these protocols that make up the TCP/IP suite are shown in the diagram below, together with the ARPA and OSI layers.

ARPA Layer	Protocol Implementation						OSI Layer
Process / Application	File Transfer	Electronic Mail	Terminal Emulation	File Transfer	Client / Server	Network Management	Application
	File Transfer Protocol (FTP) MIL-STD-1780 RFC 959	Simple Mail Transfer Protocol (SMTP) MIL-STD-1781 RFC 821	TELNET Protocol MIL-STD-1782 RFC 854	Trivial File Transfer Protocol (TFTP) RFC 783	Sun Microsystems Network File System Protocols (NFS) RFCs 1014, 1057, and 1094	Simple Network Management Protocol (SNMP) RFC 1157	Presentation
	Transmission Control Protocol (TCP) MIL-STD-1778 RFC 793			User Datagram Protocol (UDP) RFC 768			Session
Host-to-Host							Transport
Internet	Address Resolution ARP RFC 826 RARP RFC 903		Internet Protocol (IP) MIL-STD-1777 RFC 791		Internet Control Message Protocol (ICMP) RFC 792		Network
Network Interface	Network Interface Cards: Ethernet, Token Ring, ARCNET, MAN and WAN RFC 894, RFC 1042, RFC 1201 and others						Data Link
	Transmission Media: Twisted Pair, Coax, Fibre Optics, Wireless Media, etc.						Physical

Figure 4
TCP/IP Protocols

2.1 Telecommunications Network (TELNET)

TELNET provides a remote login capability. This lets a user log in to one machine from another machine (or host) and act as if he were sitting in front of that machine. The connection can be anywhere on the local network, or anywhere in the world if there is access and the user has the necessary permission to log in to the second machine.

2.2 File Transfer Protocol (FTP)

The File Transfer Protocol enables a file (or files) on one machine to be copied to another machine. The user doesn't actually log in as a full user to the machine he wants to access, as with TELNET but instead uses the FTP program to enable access. Again (as with TELNET) the correct permissions are necessary for access to the file/s.

2.3 Simple Mail Transfer Protocol (SMTP)

This protocol is used for transferring electronic mail between systems. SMTP is completely transparent to the user. SMTP works behind the scenes and users are almost never aware of the fact that SMTP is working in the background and few administrators have to bother with it. SMTP is mostly a trouble-free protocol and is in wide use today.

2.4 Kerberos

Kerberos is a widely used security protocol. It uses a special application called an authentication server to validate passwords and encryption schemes. Kerberos is one of the more secure encryption schemes used in communications and is quite common in UNIX.

2.5 Domain Name Service (DNS)

The Domain Name Service system enables a computer with a common name to be converted to a special network address. For example, a PC called Darkstar cannot be accessed by another machine on the same network (or any other connected network) unless some method of checking the local machine name and replacing the name with the

machine's hardware address is available. DNS provides a conversion from the common local name to the unique physical address of the network's connection.

2.6 Simple Network Management Protocol (SNMP)

The SNMP protocol provides status messages and problem reports can be sent across a network to an administrator. SNMP uses the User Datagram Protocol (UDP) as a transport mechanism. SNMP uses slightly different terms to the standard TCP/IP using managers and agents rather than clients and servers. An agent provides information about a device, whereas a manager communicates across a network with agents.

2.7 Network File System (NFS)

The NFS is a set of protocols developed by SUN Microsystems to enable multiple machines to access each other's drives and directories transparently. They accomplish this using a distributed file system scheme. NFS systems are common in large corporate environments, especially those that use UNIX workstations.

2.8 Remote Procedure Call (RPC)

The RPC protocol is a set of functions that enables a machine to communicate with another machine (the server). It provides for programming functions, return codes, and predefined variables to support distributed computing.

2.9 Trivial File Transfer Protocol (TFTP)

The TFTP protocol is a very simple, unsophisticated file transfer protocol that lacks security. It uses UDP for the transport mechanism. TFTP performs the same task as FTP but uses a different transport protocol.

2.10 Internet Control Message Protocol (ICMP)

The ICMP is responsible for checking and generating messages on the status of devices on a network. It can be used to inform other devices of a failure in one particular machine. ICMP and IP usually work together.

3. Routing : IP

A globally accepted method of addressing is required if any node is able to connect with any other node. All global internet addresses and the policies related to the addresses are controlled by the Internet Assigned Number Authority, and are allocated by the Internet Network Information Centre (INTERNIC). This allocation is for networks that will be attached to the worldwide Internet. For those networks that will never connect to the Internet, unique addresses can be allocated by the local corporation network administrator who is responsible for maintaining the local TCP/IP network. The TCP/IP approach with addressing is to assign each node on the internet a unique 32 bit internet address called its internet address.

The structure of the internet address is shown below:

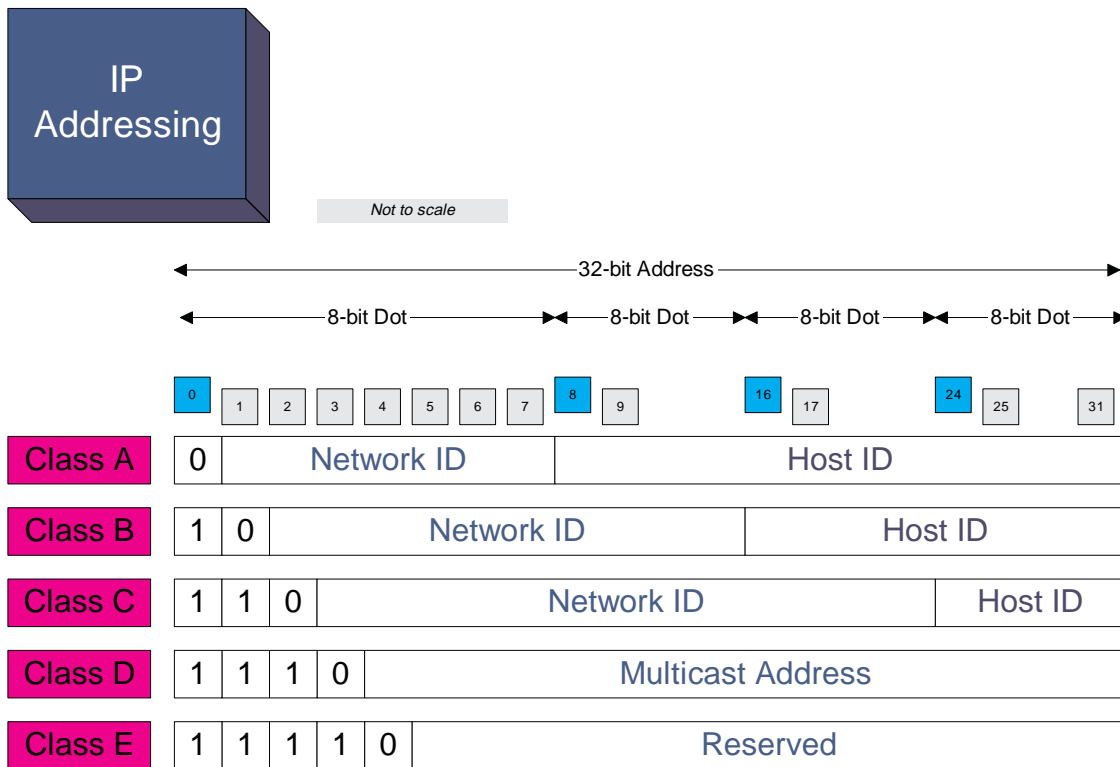


Figure 5
TCP/IP Addressing Structure

Each address consists of the following components:

- Network ID the network number
- Host ID the address of the host or node on the network

Generally the first three classes are used. Class A addresses have a Network ID of 7 bits and Host ID of 24 bits and are for the small number of networks which have up to 16,777,214 nodes. Class B addresses are for medium size networks with up to 65,534 nodes and Class C networks have less than 254 nodes.

3.1 Routing in an Internet

Routing defines the activity of selecting the path over which to send the packets. Router refers to any computer making the decision on which path or route to use.

Both nodes and gateways are involved in the routing of the IP message. There are two forms of routing:

- **Direct Routing**

Where a node transfers a message directly to another, all connected on the same network.

- **Indirect Routing**

Where the destination node is not directly connected and the message has to be passed to a gateway for delivery.

For direct routing on a particular physical network, the transmitting node encapsulates the datagram in a physical frame (such as Ethernet), binds the destination internet address to the physical hardware address and physically transmits the message. The transmitter easily knows whether the transfer problem is direct routing as it compares the network portion of the destination IP address with its own network identification. This is quickly done within the software.

Indirect routing is done by software algorithms (and data tables) in each gateway examining the datagram's destination network address (as it arrives) and then forwarding the datagram to the next appropriate gateway (in the direction of the destination node). Note that the gateway routing software is only concerned with the network address - the node address (and indeed the host part of the IP address) is not relevant.

The technique used for efficient routing of the datagram is to use an internet routing table which stores information on all destination networks and how to reach them. The routing table can only point to gateways that are physically connected to the network to which the gateway (which currently has the datagram) is connected. The size of the routing table depends on the number of interconnected networks. The number of individual nodes connected to the various networks does not impact on the size of the routing table.

3.2 Routing and the Physical Address

When the routing algorithm extracts the next hop address (where the datagram must be sent to next in its path), it passes this information and the datagram to the physical network software which binds the next hop address to a physical address. The physical network software then forms a frame using that physical address as the destination address (for this one hop only), and places that datagram in the data section of the message and despatches the packet. The reason for the IP routing software not using or calculating the physical addresses is to isolate the IP software from the physical details of the network (and hide the underlying details of the network from the IP software).

3.3 Field Destination

When the datagram arrives at a destination host (or node), the physical network interface software transfers it to the IP software for further processing. There are two cases:

- **Match**

The datagram IP destination address matches that of the destination node and the message is processed by the IP software.

- **No Match**

The datagram destination IP address does not match and the node is then required to get rid of the datagram (and not to forward it to another node).

3.4 IPv6

The Internet Protocol – of any version – is the fundamental platform upon which the entire Internet Protocol suite is based. Any change to the IP affects many other areas of the suite, and there are over sixteen recent RFCs that have been published to this effect. The changes that have been proposed for the change from Ipv4 to IPv6 include:

- an address size of 128 bits (over the 32 bit IPv4 size)
- having optional header fields
- better support for extensions and options
- the ability to label the flow of data
- security, including privacy and authentication changes

The diagram below illustrates the IPv6 header format.

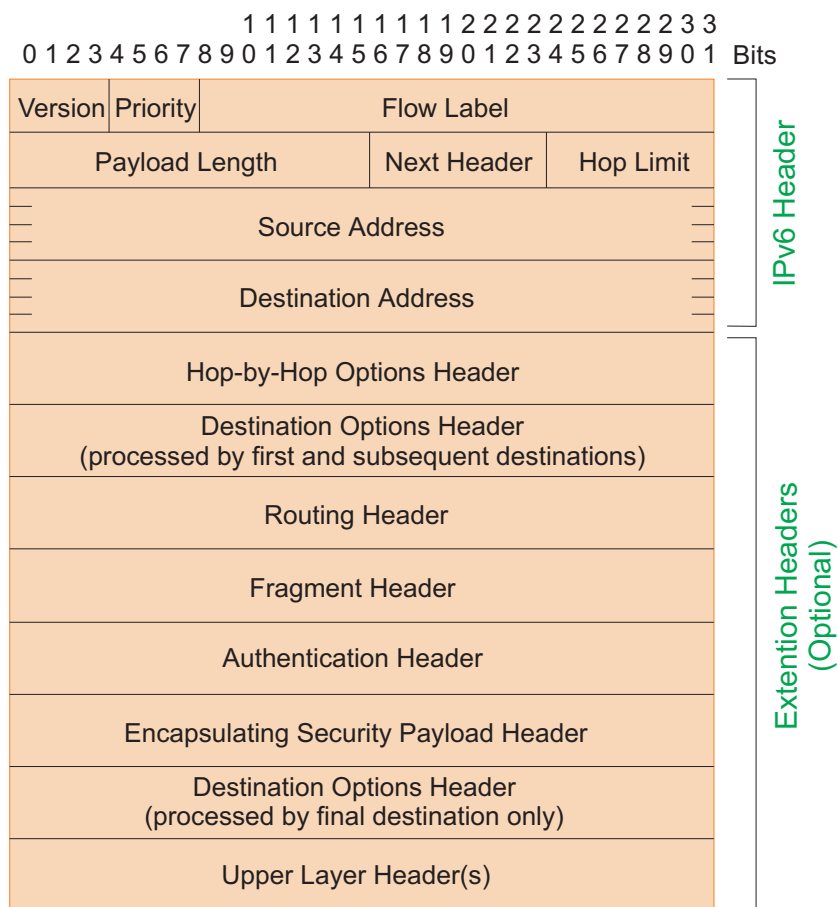


Figure 6
IPv6 Packet Format

4. Network Interface Connection

The Network Interface Connection refers to the physical layer. This is the connection between the ARPA layers of TCP/IP and the network being used for data transfer which may be a local area network (LAN), a metropolitan area network (MAN), or a wide area network (WAN).

The Network Interface Layer must reside in all hosts (machines connected to the network) but the actual type of network that is used in different segments of the network (another LAN) may be of a different type. For example, in the illustration below, the Host A machine may be on an Ethernet network connected to the Router B, while Host Z might be on a Token Ring network.

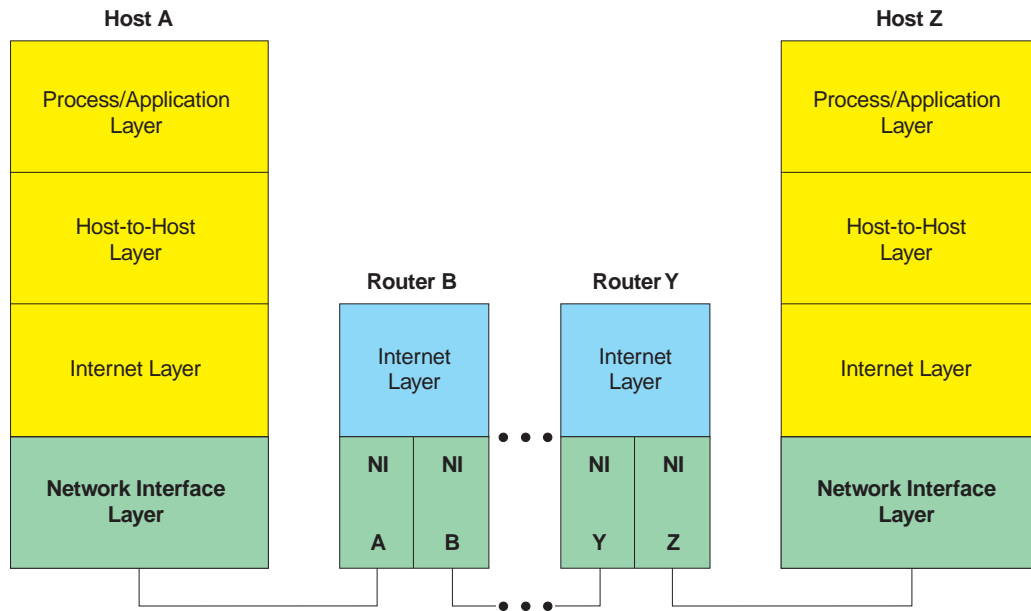


Figure 7

The Network Interface Connection

The next figure shows the Network Interface Layer RFCs and options. The TCP/IP frame that is transmitted is contained within the network frame.

ARPA Layer	Protocol Implementation						OSI Layer
Process / Application	File Transfer	Electronic Mail	Terminal Emulation	File Transfer	Client / Server	Network Management	Application
	File Transfer Protocol (FTP) MIL-STD-1780 RFC 959	Simple Mail Transfer Protocol (SMTP) MIL-STD-1781 RFC 821	TELNET Protocol MIL-STD-1782 RFC 854	Trivial File Transfer Protocol (TFTP) RFC 783	Sun Microsystems Network File System Protocols (NFS) RFCs 1014, 1057, and 1094	Simple Network Management Protocol (SNMP) RFC 1157	Presentation
	Transmission Control Protocol (TCP) MIL-STD-1778 RFC 793			User Datagram Protocol (UDP) RFC 768			Session
Host-to-Host	Address Resolution ARP RFC 826 RARP RFC 903		Internet Protocol (IP) MIL-STD-1777 RFC 791	Internet Control Message Protocol (ICMP) RFC 792		Transport	
Internet	Network Interface Cards: Ethernet, Token Ring, ARCNET, MAN and WAN RFC 894, RFC 1042, RFC 1201 and others						Network
Network Interface	Transmission Media:						Data Link
	Twisted Pair, Coax, Fibre Optics, Wireless Media, etc.						Physical

Figure 8

ARPA Network Interface Layer Protocols

4.1 Ethernet

The current version of Ethernet used widely throughout the networking community is Ethernet Version 2. This version transmits data at **10Mbps** and uses **48-bit** addressing. The Ethernet frame format defines a length between **64** and **1,518** octets including the header, trailer and data. The header must contain the source and destination addresses (each 6 octets – or 48 bits – in length), and a 2 octet “**type**” field. The TCP/IP “address” protocols (**ARP** and **RARP**) perform address mapping between the TCP/IP 32-bit address and the Ethernet 48-bit address. These protocols are covered in detail in the IDC TCP/IP course.

The actual data in an Ethernet frame must be between 46 and 1,500 octets in length. IP will pad extremely short datagrams with zeros if necessary to increase the size to the minimum 46 octets. The maximum length of the Ethernet data field is 1,500 octets, while the maximum Transmission Unit allowed by IP is 576 octets.

4.2 Frame Relay Example

In the example below, a user wants to access a file that resides on the server at the main office. A frame relay network connects the sites using two permanent virtual connections (PVCs). The PVCs operate over a 56Kbps leased line. The IP traffic between the machines is carried on the PVC using Data Link Connection Identified (DLCI) 140. The Network Analyser is positioned near the server, and a careful note should be made of the addresses used – source and destination. In the frame sequences listed, the destination for frame 1078 is the server (DTE), while the source is the DCE (network) which actually refers to the host machine sending data through the network. The host initiates a file transfer process and indicates that an ASCII tye transfer is required. The server confirms the ASCII type transfer in frame 1079, and frames 1082 and 1083 confirm the ports which will be used for the transfer. In frame 1088 the host advises the name of the file being requested, and the server opens a data connection sequence and begins the file transfer in frame 1093. Finally the server completes the transfer in frame 1107.

<i>Frame</i>	<i>Destination</i>	<i>Source</i>	<i>Summary</i>
1078	DTE	DCE	FTP C PORT=1214 TYPE A<0D0A>
1079	DCE	DTE	FTP R PORT=1214 200 Type set to A<0D0A>
1082	DTE	DCE	FTP C PORT=1214 PORT 161,69,133,72,4,254<0D0A>
1083	DCE	DTE	FTP R PORT=1214 PORT 200 Port command successful<0D0A>
1088	DTE	DCE	FTP C PORT=1214 PORT RETR MyFile.txt<0D0A>
1092	DCE	DTE	FTP R PORT=1214 150 Opening ASCII mode data connection for MyFile.txt<0D0A>
1107	DCE	DTE	FTP R PORT=1214 226 Transfer complete<0D0A>

4.3 Troubleshooting the Network Interface

A few suggestions here are:

- Check the basic communications path between stations and the network for problems such as broken cables and loose connectors.
- Check for compliance with appropriate hardware standard (e.g. Ethernet V.2 against Ethernet 802.3)

5. Internet Connection

The Internet layer in the ARPA model is responsible for delivery of a datagram from the source to the destination address. A common problem that needs to be resolved here is the incorrect assignment of IP addresses and subnet masks.

Subnetworks are used for forming an hierarchical routing structure within a series of interconnected networks. It works simply as follows:

Each PC on a network is configured with a particular subnetwork mask. As discussed earlier, each address comprises a network and a host (or PC) portion. If there are multiple physical networks, the host portion of the address can be further broken down into a subnet portion and a host address indicating a particular device on the network.

The subnet mask is used by the host machine to make a routing decision. A subnet mask is a 32-bit number that has one's in the Network ID and subnetwork ID portion. Zero's are in the Host ID field. A logical AND function is performed between the Subnet Mask and the destination IP address. This result is compared to an AND function performed on the Subnet Mask and the source IP address. If the results are the same; the source and destination IP address belong to the same network. If the results are different, the two addresses are on different subnetworks and the datagram must go to a router for delivery.

This is why it is important to choose the subnet mask correctly; otherwise the message may be sent off to a router on the network erroneously; which means it will never get sent to the correct destination address.

Typical suggestions for troubleshooting at this layer are:

- Ensure IP addressing and subnet masks are correctly set up. Use of the PING utility is often useful to identify problems here.
- For more complex problems, use of the TRACEROUTE utility is useful in tracing the route the packet follows through an interconnected set of networks.

6. Host to Host Connection

By “*host to host connection*” we are talking about the protocol being used for data transmission. There are two protocols used for this : User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). The host layer reside on the hosts communicating across an internet, and does not reside in the routers.

The host may be dealing with more than one process, and therefore it is important that the datagram that is transmitted include some information about the host access point – called a **port**. Another issue of concern is for the host to know whether the datagram arrived correctly, and part of the header includes information to ensure this. Another issue concerns the overhead used for the transmission. UDP has much less overhead than TCP. The position of the host protocols in the ARPA model is shown below.

Figure below illustrates where the overhead for the type of transmission protocol being used occurs. The UDP header requires a minimum of 8 octets, whereas the TCP header uses a minimum of 20 octets in its header information.

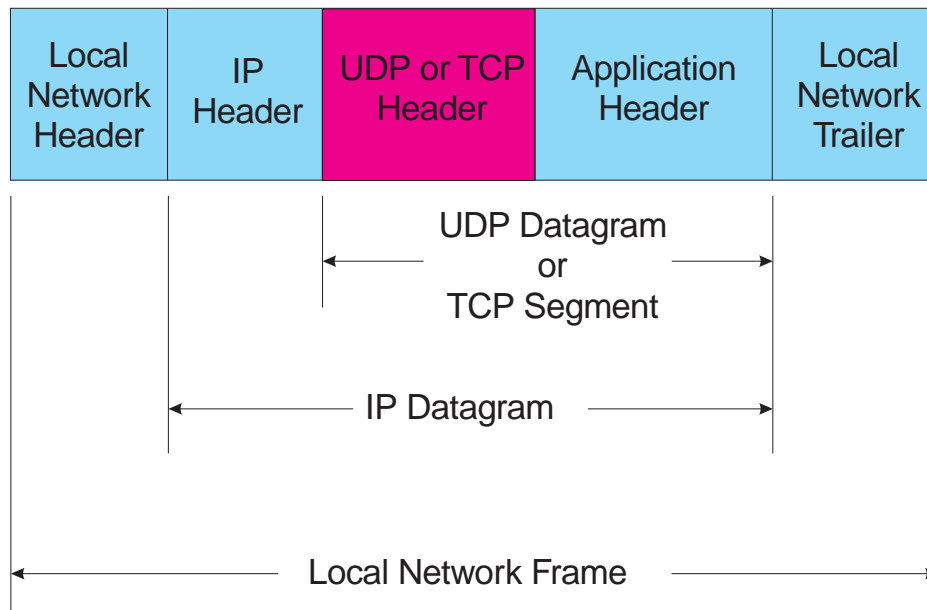


Figure 9
UDP/TCP Header Position

On top of hardware addressing and IP addressing comes the host “addressing” mechanism, or **port addresses**. Port numbers are necessary because TCP/IP assumes (quite rightly) that there may be several host applications running on a machine which require different access to a host data transfer protocol. Port numbers are 16 bits long and are standardised according to their use.

Some common port numbers are illustrated below:

<i>Port No</i>	<i>Description</i>
7	Echo
20	ftp-data
21	ftp (control)
22	TELNET
23	World Wide Web HTTP

6.1 Transmission Control Protocol (TCP)

Although TCP is generally considered part of the TCP/IP internet protocol suite, it is a protocol that can be used independently. It can be considered to operate independently of the underlying physical network.

The Transmission Control Protocol (TCP) specifies the structure of the messages, the acknowledgments between two nodes for reliable data transfer, how messages are routed to multiple destinations on a machine and how errors are detected and corrected.

TCP is quite efficient although it is written as a general application. It can run at 8 Mbps for two workstations operating on a 10 Mbps link or - with a super computer – can operate at up to 600 Mbps.

There is a need for a reliable means of delivering message packets. Packets can be lost or destroyed with transmission errors, or when networks become too heavily loaded to cope with an overloaded situation. Application programs at the highest level need to be able to send data from one point to another in a reliable manner. It is important to have a general purpose protocol such as TCP to handle the problems of reliable transmission of data which the application programs can then use.

6.2 User Datagram Protocol

The User Datagram protocol provides a connectionless host-to-host communication path. UDP assumes that IP is the underlying routing protocol. As UDP is connectionless, there is a much smaller header associated with UDP than with TCP. The first two fields of the UDP header are the source and destination addresses. The source field is optional, and when not used is filled with zeros. The length field for a UDP datagram is 2 octets in length, and specifies the length of the UDP datagram (with a minimum of 8 octets). While the IP address is used to route the UDP datagram to the correct IP address (host) on the network, a UDP Port address routes the datagram to the correct host process. UDP therefore, adds port addressing capabilities to the IP addressing feature. An example of a protocol that uses UDP would be the “Time” protocol. Other protocols that use the UDP transport mechanism include DNS, BOOTP, TFTP and RPC. All of these protocols assume that if there is a transfer failure, that the higher level protocol will assume responsibility for correcting the problem.

6.3 Troubleshooting the Host to Host connection

A few suggestions here are:

- Determine whether the underlying protocol – UDP or TCP is adequate for the application.
- With TCP check that the correct port number has been assigned and look for other events in the connection such as establishment of the connection and termination.

7. Process and Application

The Process layer sits at the very top of the ARPA architectural model. This layer may be accessed directly by users interacting with applications such as TFTP, FTP, TELNET and SNMP.

The next two figures show the layers and applications at this part of the ARPA model.

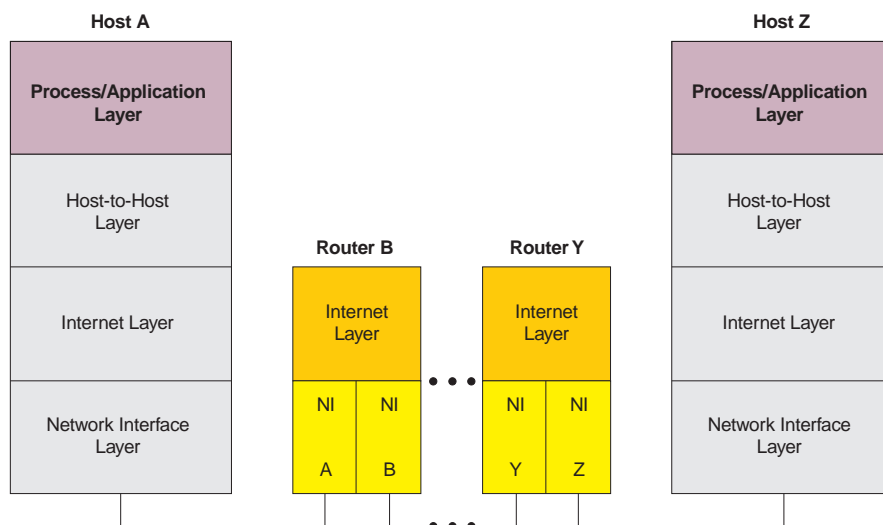


Figure 10

The Process and Application Protocols

FTP allows local and remote client and server machines to share files and data using the reliable transport mechanism provided by TCP. A server machine running an FTP server service includes a Server Data Transfer Service and the Server Protocol Interpreter. On the client side, the FTP client runs three services including a client User Data Transfer

Process, a User Protocol Interpreter, and a User Interface – through which the user interacts with the application. Two port numbers are used for FTP : Port number 21 is assigned to the FTP control line, and Port number 20 is assigned to FTP data. Proper operation of the data transfer between the User and Server Data Transfer Processes depends on the control commands that go between the User and Server Protocol Interpreters. The commands are in three categories including Access Control, Transfer Parameters and Service. The commands are listed in detail in the IDC TCP/IP course manual. RFC 959 details the reply messages from FTP commands.

A typical FTP scenario begins with a command to initiate a connection. This would take the form of:

FTP [host address]

The host responds asking for a username and password login.

The user would then navigate to the correct directory that contains the file/s being copied using the **cd** command.

The “**type**” of file transfer should then be specified (binary, text, etc).

The user would then issue a command to get the file, along the lines of **get filename.ext**.

The FTP **quit** command ends the session.

7.1 SNMP Example

An Agent/Manager carries network management information between devices on a distributed managed network. The type and amount of network management information transmitted over the internet determines the amount of processing and storage the Agent and the Manager will require, and it may also impact the choice of network management protocol. There are two choices for the management protocol, SNMP for more simple management, and CMIP/CMOT for complex management.

The management system contains many components including the Structure of Management Information (**SMI**), the Management Information Base (**MIB**) and the protocol being used (eg **SNMP**).

There are two versions of SNMP. Version 1 (SNMP1) is widely supported and is included in most networking products. SNMP2 has many improvements over SNMP1, which will be covered in the IDC TCP/IP course. With the large volume of data contained in large networks – with every device being managed – there is a need to manage that information . The Structure of Management Information (**SMI**) provides the means to do this. **SMI** gives a method for naming and organising objects, while the **MIB** stores information about each object.

The SMI uses a conceptual tree with various objects representing the leaves of the tree. The objects are represented using the Abstract Syntax Notation ISO protocol concepts. The SMI assigns each object a sequence of integers, known as an Object Identifier. Refer to the figure below. I

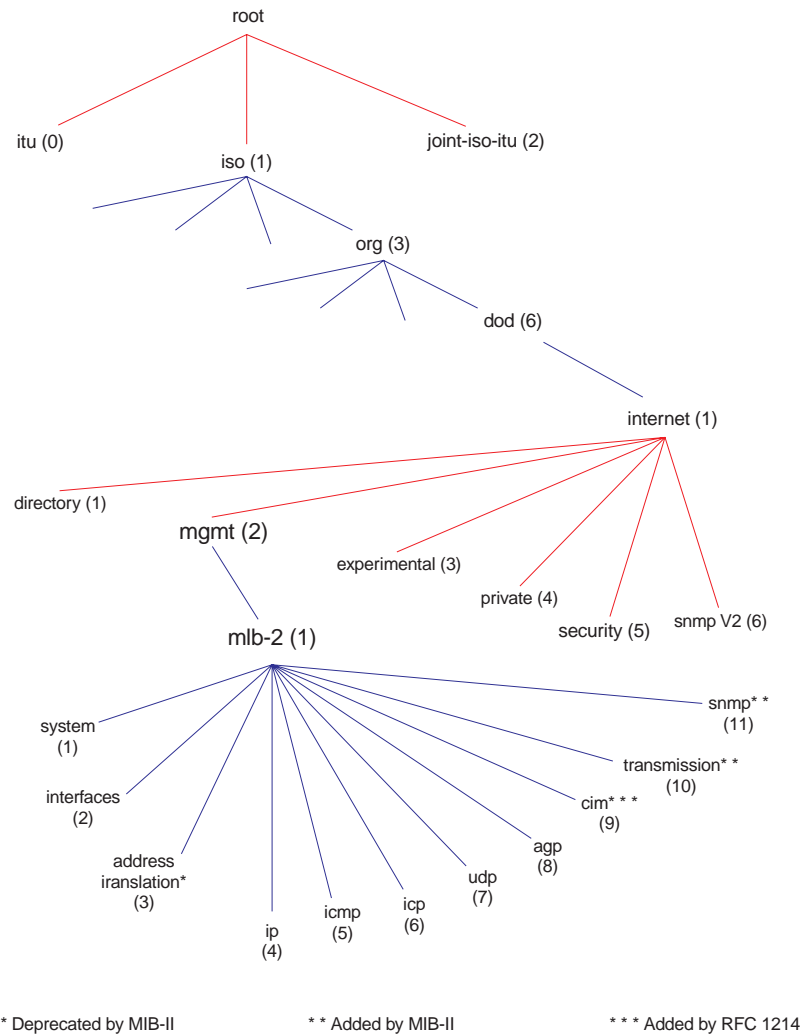


Figure 11
The Internet Management Information Base Tree

7.2 Troubleshooting the Process/Application Layer

A few pointers here are:

- Is data being interpreted correctly at both ends?
- Check for any incompatibilities in internal implementation of the application layer between the two systems communicating.