INTERNETWORKING WITH TCP/IP 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Int : **40** Ext : **60** Total: **100**

SCOPE

Build and understanding of the fundamental concepts of computer networking. Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

OBJECTIVES

- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparting the student for entry advanced courses in computer networking.
- Understand various transmission media, their comparitive study, fiber optics and wireless media.

UNIT I

Introduction: History of Internet – Interconnecting devices – protocols and standards – TCP/IP protocol suite – internetworking devices – routing concept – classfull IP addressing – subnetting – supernetting – classless addressing.

UNIT II

ARP & RARP – Proxy ARP – ARP over ATM – ARP and RAPP protocol format. IP datagram – Fragmentation – Options – IP datagram format – Routing IP datagrams – Cheksum. ICMP – types of messages – message format – Error reporting – Query – Cheksum.

UNIT III

Group management – IGMP message – IGMP operation – process to process communication – UDP operation – TCP services – Flow control.

UNIT IV

BOOTP – DHCP – Address discovery and Binding. DNS – Name Space – DNS in Internet – Resolution – Resource Records.

UNIT V

Remote Login- FTP – SMTP – SNMP . IP over ATM Wan – Cells – Routing the Cells – ATMARP – Logical IP SUBNETS – VPN

Suggested Readings

1. Behrouz A. Forouzan (2010). TCP/IP protocol suite (4th ed.). New Delhi: Tata McGraw Hill publication.
2. Andrews S.Tanenbaum (2003). Computer Networks. (4th ed.). New Delhi: Prentice Hall of India Private Ltd.
3. Buck Graham (2007). TCP/IP addressing (2nd ed.). New Delhi: Harcount India Private Limited.
4. Douglas E. Comer (2000). Computer Networks and Internets (4th ed.). New Delhi: Pearson education Asia.
5. William Stallings (2007). Data and Computer Communication Network (8th ed.). New delhi: Tata McGraw Hill

Web sites

1. https://www.tutorialspoint.com
2. https://www.geeksforgeeks.org
3. https://electronicspost.com/ip-datagram-fragmentation-with-example/
4. https://networklessons.com
5. http://sourcedaddy.com/networking/
6. https://en.wikipedia.org/wiki
7. https://technet.microsoft.com

ESE Pattern		
1	Part - A	20 X 1 = 20 Marks
2	Part - B	5 X 2 = 10 Marks
3	Part – C	5 X 6 = 30 Marks
4	**Total**	**60 Marks**

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
Established Under Section 3 of UGC Act, 1956)
Coimbatore – 641021, INDIA
Department of Computer Science, Applications & Information Technology

Subject Name: Internetworking with TCP/IP **Subject Code: 17CTU403**

Semester: IV **Class: II Bsc. CT**

Staff: B.Bharathi

S.No	Topics	No. of Periods Required	Reference Materials
	Unit – I		
1	Introduction: History of Internet	1 hr	SR1: 1- 6
2	Connecting Devices	1 hr	SR1: 69 - 74
3	Protocols and Standards, TCP/IP protocol suite	1 hr	SR1: 6-7, 30-33, W1
4	Internetworking Devices, Routing Concept	1 hr	SR1: 69 – 74, W2
5	Classful IP Addressing	1 hr	SR1: 84 - 95
6	Subnetting , Supernetting	1 hr	SR1: 102 - 109
7	Classless Addressing	1 hr	SR1: 115 - 124
8	Recapitulation of Important Topics	1 hr	-
	Total Number of Hours	**8 hrs**	
Suggested Readings **SR1:** Behrouz A. Forouzan (2010). TCP/IP protocol suite (4th ed.). New Delhi: Tata McGraw Hill publication.			
Web References **W1:** https://www.tutorialspoint.com **W2:** https://www.geeksforgeeks.org/network-devices-hub-repeater-bridge-switch-router-gateways/			
Unit – II			
1	ARP & RARP, Proxy ARP	1 hr	SR1: 160 – 166, W2
2	ARP over ATM, ARP and RARP Protocol Format	1 hr	SR1: 166 - 173
3	IP Datagram, Fragmentation	1 hr	SR1 : 180 – 188, W1
4	Options, IP Datagram Format	1 hr	SR1: 188 – 200, W2

5	Routing IP Datagrams, Checksum	1 hr	R4 : 200
6	ICMP, Message Format, Types of Messages, Error Reporting	1 hr	SR1: 212 – 221, W2
7	Query, Checksum	1 hr	SR1: 221 - 226
8	Recapitulation of Important Topics	1 hr	-
	Total Number of Hours	**8 hrs**	

Suggested Readings

SR1: Behrouz A. Forouzan (2010). TCP/IP protocol suite (4th ed.). New Delhi: Tata McGraw Hill publication.

Web References

W1: https://electronicspost.com/ip-datagram-fragmentation-with-example/

W2: https://www.geeksforgeeks.org

Unit - III

1	IGMP Message	1 hr	SR1: 237 – 239, W1
2	IGMP Operation	1 hr	SR1: 239 - 244
3	Process to Process Communication: Port Number	1 hr	SR1: 256 - 260
4	User Datagram & Operations • Connection less Service • Flow and Error Control • Encapsulation, Decapsulation • Multiplexing	1 hr	SR1: 260 – 267, W1
5	TCP Services, Features	1 hr	SR1: 276 – 284, W1
6	Flow Control: Sliding Windows & Syndrome	1 hr	SR1: 299 – 305, W3
7	Recapitulation of Important Topics	1 hr	-
	Total Number of Hours	**7 hrs**	

Suggested Readings

SR1: Behrouz A. Forouzan (2010). TCP/IP protocol suite (4th ed.). New Delhi: Tata McGraw Hill publication.

Web References

W1: https://networklessons.com/tag/igmp/

W2: https://www.tutorialspoint.com

W3: https://www.geeksforgeeks.org

Unit - IV			
1	BOOTP	1 hr	SR1: 457 – 461, W1
2	DHCP: Dynamic address Allocation, manual Configuration, Packet Format	1 hr	SR1: 463 – 467, W3
3	Address Discovery and Binding	1 hr	W2
4	DNS, Name Space , Distribution of Name Space	1 hr	SR1: 472 – 477, W2
5	DNS in Internet	1 hr	SR1: 471 - 481
6	Resolution & Resource Record	1 hr	SR1: 481 – 482, 486 - 488
7	Recapitulation of Important Topics	1 hr	
	Total Number of Hours	**7 hrs**	-
Suggested Readings **SR1:** Behrouz A. Forouzan (2010). TCP/IP protocol suite (4th ed.). New Delhi: Tata McGraw Hill publication.			
Web References **W1:** http://sourcedaddy.com/networking/what-about-bootp.html **W2:** https://www.tutorialspoint.com **W3:** https://www.geeksforgeeks.org			
Unit – V			
1	Remote Login	1 hr	SR1: 610 – 630, W1
2	FTP	1 hr	SR1: 630 – 643, W3
3	SMTP	1 hr	SR5: 824 - 840
4	SNMP, Management Components	1 hr	SR5: 877 – 897, W3
5	IP over ATM Wan, Cells, Routing the Cells	1 hr	SR1: 621 - 627
6	ATMARP, Logical IP	1 hr	SR1: 228 - 232
7	VPN	1 hr	SR1:680 – 684, W2
8	Recapitulation of Important Topics	1 hr	-
9	Discussion of Previous ESE Question Papers	1 hr	-
10	Discussion of Previous ESE Question Papers	1 hr	-
	Total Number of Hours	**10**	
	Total No. of Periods all Units (8+8+7+7+10)	**40 hrs**	

Suggested Readings
SR1: Behrouz A. Forouzan (2010). TCP/IP protocol suite (4th ed.). New Delhi: Tata McGraw Hill publication.
SR5: William Stallings (2007). Data and Computer Communication Network (8th ed.). New delhi: Tata McGraw Hill publication.
Web References:
W1: https://en.wikipedia.org/wiki/Telnet
W2: https://technet.microsoft.com
W3: https://www.geeksforgeeks.org

Suggested Readings

1. Behrouz A. Forouzan (2010). TCP/IP protocol suite (4th ed.). New Delhi: Tata McGraw Hill publication.
2. Andrews S.Tanenbaum (2003). Computer Networks. (4th ed.). New Delhi: Prentice Hall of India Private Ltd.
3. Buck Graham (2007). TCP/IP addressing (2nd ed.). New Delhi: Harcount India Private Limited.
4. Douglas E. Comer (2000). Computer Networks and Internets (4th ed.). New Delhi: Pearson education Asia.
5. William Stallings (2007). Data and Computer Communication Network (8th ed.). New delhi: Tata McGraw Hill

Web sites

1. https://www.tutorialspoint.com
2. https://www.geeksforgeeks.org
3. https://electronicspost.com/ip-datagram-fragmentation-with-example/
4. https://networklessons.com
5. http://sourcedaddy.com/networking/
6. https://en.wikipedia.org/wiki
7. https://technet.microsoft.com

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed University Established Under Section 3 of UGC Act, 1956)
Coimbatore – 641021, INDIA
Department of Computer Science, Applications & Information Technology

Unit - I

S.No	Question	Choice1	Choice2	Choice3	Choice4	Answer
1	_____ is composed of hundreds of thousands of interconnected networks.	Internet	Intranet	Extranet	Arpanet	Internet
2	ARPA Stands for _____	Advanced Research Protocol Agency	Automated Research Provider Agency	Advanced Research Project Agency	None	Advanced Research Project Agency
3	Two types of ARPANET are _____	NSFNET & CSNET	ANSNET & MILNET	INTERNET & INTRANET	ARPANET & MILNET	ARPANET & MILNET
4	In ARPANET each host is attached to a specialized computer called _____	MIP	IMP	PIM	IGMP	IMP
5	___ is responsible for higher level function such as error detection	IP	TCP/IP	NCP	TCP	TCP
6	IP handles _____	Segmentation	Error Detection	Datagram Routing	Reassembly	Datagram Routing
7	_____ is a set of rule that governs data communication	Standards	Protocols	Organization	Routing	Protocols
8	Key elements of Protocols are _____	Defacto & Dejure	Requirement & Packet size	Interface & Timing & Packets	Syntax & Semantic & timing.	Syntax & Semantic & timing.
9	_____ Standard that have not been approved by the Organized body	Dejure	Defacto	Dejery	None	Defacto
10	_____ is the Organization and _____ is the Model	ISO , OSI	OSI , ISI	ISA , OSI	ISA , ISI	ISO , OSI
11	OSI Stands for _____	Open System Interconnection	Open Standard Interconnection	Organizational Standard interface	Open Source interconnection	Open System Interconnection
12	The Process on each machine that communicate at a given layer is _____	Interface	Point – to - Point	Routing	Peer to Peer	Point – to - Point
13	_____ Layer is responsible for delivery of a message from one process to another	Transport layer	Data Link layer	Physical layer	Network layer	Transport layer
14	_____ provides services to the user	Application Layer	Transport Layer	Session Layer	Presentation Layer	Application Layer
15	In TCP/IP application layer is the combination of ____,_____ and ___	Application , Network, Data Link	Application , Network, Physical	Application , Network, Session	Application Layer	Application Layer
16	Which of the following is not a connection device _____	Hub	Router	Amplifier	Bridge	Amplifier
17	Repeater is a _____ not a _____	Amplifier, Regenerator	Regenerator , Amplifier	Connector, segmented	None	Regenerator , Amplifier
18	_____ is a multi port repeater	Bridge	Router	Hub	None	Hub
19	_____ has filtering capacity	Router	Hub	Bridge	Both b & c	Bridge
20	Router is a _____ device.	One layer	Two layer	Tree layer	Four Layer.	Tree layer
21	An IP Address is a _____address	4 byte	8 byte	34byte	1 byte	4 byte
22	In_____ notation one or more spaces is inserted between each octet.	Hexadecimal	Binary	ASCII	Decimal	Binary
23	Each Octet is referred to as a _____	Bit	Byte	Word	Pixel	Byte
24	In Class full addressing IP address is divided into _____ Classes	3	5	4	6	5
25	if the first two its are zero, then the IP address is of _____ Class	A	B	C	D	A
26	The range of Class D address is _____	223 to 240	224 to 239	225 to 237	221 to 234	224 to 239
27	In _____ the station are unaware of the existence of bridge	Transformer	Transparent Bridge	Bridge	None	Transparent Bridge
28	ARP stands for_____	Address Reverse Protocol	Address Resolution Protocol	Advanced Research Project	Advanced Resolution Protocol	Address Resolution Protocol
29	The protocol used to associate an IP address with physical address is ___	ARP	RARP	PROXY ARP	ICMP	RARP
30	_____is an internet work address	physical address	Logical address	IP address	Network address.	Logical address
31	The logical address in the TCP/IP protocol suit are called_____	logical address	Network address	IP address	Physical address	IP address
32	In _____each time a machine knows one of the 2 addresses.	Dynamic mapping	Static mapping	Temporary mapping	None of the above.	Dynamic mapping
33	RARP Stand for_____	Resolution Address Reverse Protocol	Routing Address Resolution Protocol	Routing Address Reverse protocol	Reverse Address Resolution Protocol	Reverse Address Resolution Protocol
34	____ allows a host to discover its internet address when it knows only its physical address.	ARP	PROXY ARP	RARP	ICMP	RARP
35	_____ means creating a table that associates a logical address with the physical address.	Dynamic mapping	Static mapping	Temporary mapping	None of the above.	Static mapping
36	__ is a 16-bit field defining the type of the network on which ARP is running.	Protocol type	Network type	Software type	Hardware type.	Hardware type.
37	_____is a m16-bit field defining the protocol.	Hardware type	Software type	Protocol type	Network type.	Protocol type
38	An ARP request is_____	Unicast	Broadcast	Telecast	None of the above.	Broadcast
39	An ARP reply is_____	Broadcast	Telecast	Unicast	None of the above.	Unicast
40	_____is used to create a subnetting effect.	PROXY ARP	ARP	RARP	ICMP	PROXY ARP
41	An ARP package involves_____components.	2	8	5	7	5
42	_____ Waits until an ARP Packet arrives.	Output module	Input module	Control module	None of the above.	Input module
43	Packets in the IP Layer are called_____	Components	Interface	Tokens	Datagram.	Datagram.
44	The process of dividing the datagram to pass through the tworks_____	Segmentation	Fragmentation	Splitting	Encapsulation	Fragmentation
45	The two-bit subfield defines the general purpose of the option is _____	Copy	Class	Number	object	Class
46	The error detection method used by most TCP/IP protocol is called the _____	checksum	parity	checkerror	None of the above	checksum
47	The maximum length of the diagram is _____ bytes	65,534	65,535	66,334	65,432	65,535
48	A datagram consist of _____ and _____	Title and data	Header and information	Content and header	Header and data	Header and data

49	ICMP stands for _____	Internet Control Message Protocol	Intranet Control Message Protocol	Internet Content Message Protocol	Intranet Content Message Protocol	Internet Control Message Protocol
50	ICMP is a _____ layer protocol	Application	Transport	Network	Physical	Network
51	ICMP messages are divided into _____ categories.	5	2	3	4	2
52	One of the main responsibilities of ICMP is to _____ errors.	Detect	Correct	Check	Report	Report
53	An ICMP message has _____ byte header.	8	6	4	2	8
54	There is no _____ mechanism in the IP protocol.	Data Control	Source Control	Flow Control	Error Control	Flow Control
55	A _____ message is send from router to a host on the same local network.	Redirection	Bidirection	Unidirection	Multidirection	Redirection
56	_____ program is used to find if a host is alive and responding.	Traceroute	Ping	Tracert	None of the above.	Ping
57	A parameter problem message can be created by a _____	Router	hub	Bridge	Both a and b	Router
58	_____table is used by the reassembly module.	Fragmentation table	Bridge table	Reassembly table	Routing table	Reassembly table
59	A _____ option is used to record the time of datagram processing.	Loose source Route	Time Stamp	Time Slicing	check sum	Time Stamp
60	An end of option is a _____ byte option.	4	6	8	1	1

SYLLABUS

Introduction: History of Internet – Interconnecting devices – protocols and standards – TCP/IP protocol suite – internetworking devices – routing concept – classful IP addressing – subnetting – supernetting – classless addressing

1. History of Internet

A network is a group of connected, communicating devices such as computers and printers. An internet (note the lowercase i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase I)

ARPANET

In the mid-1960s, mainframe computers in research organizations were stand-alone devices. Computers from different manufacturers were unable to communicate with one another. The Advanced Research Projects Agency (ARPA) in the Department of Defense (DOD) was interested in finding a way to connect computers together so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an interface message processor (IMP). The IMPs, in turn, would be connected to each other. Each IMP had to be able to communicate with other IMPs as well as with its own attached host.

By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the Network Control Protocol (NCP) provided communication between the hosts.

Birth of the Internet

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the Internetting Project. They wanted to link different networks together so that a host on one network could communicate with a host on a second, different network.

There were many problems to overcome: diverse packet sizes, diverse interfaces, and diverse transmission rates, as well as differing reliability requirements. Cerf and Kahn devised the idea of a device called a gateway to serve as the intermediary hardware to transfer data from one network to another.

Transmission Control Protocol/Internetworking Protocol (TCP/IP)

Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of data. This was a new version of NCP. This paper on transmission control protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. A radical idea was the transfer of responsibility for error correction from the IMP to the host machine. This ARPA Internet now became the focus of the communication effort. Around this time responsibility for the ARPANET was handed over to the Defense Communication Agency (DCA). In October 1977, an internet consisting of three different networks (ARPANET, packet radio, and packet satellite) was successfully demonstrated.

Communication between networks was now possible. Shortly thereafter, authorities made a decision to split TCP into two protocols: Transmission Control Protocol (TCP) and Internet Protocol (IP). IP would handle datagram routing while TCP would be responsible for higher level functions such as segmentation, reassembly, and error detection. The new combination became known as TCP/IP. In 1981, under a DARPA contract, UC Berkeley modified the UNIX operating system to include TCP/IP. This inclusion of network software along with a popular operating system did much for the popularity of networking.

The open (non-manufacturer-specific) implementation on Berkeley UNIX gave every manufacturer a working code base on which they could build their products. In 1983, authorities abolished the original ARPANET protocols, and TCP/IP became the official protocol for the ARPANET. Those who wanted to use the Internet to access a computer on a different network had to be running TCP/IP.

MILNET

In 1983, ARPANET split into two networks: MILNET for military users and ARPANET for nonmilitary users.

CSNET

Another milestone in Internet history was the creation of CSNET in 1981. CSNET was a network sponsored by the National Science Foundation (NSF). The network was conceived by universities that were ineligible to join ARPANET due to an absence of defense ties to DARPA. CSNET was a less expensive network; there were no redundant links and the transmission rate was slower. It featured connections to ARPANET and Telenet, the first commercial packet data service. By the middle 1980s, most U.S. universities with computer science departments were part of CSNET. Other institutions and companies were also forming their own networks and using TCP/IP to interconnect. The term Internet,

originally associated with government-funded connected networks, now referred to the connected networks using TCP/IP protocols.

NSFNET

With the success of CSNET, the NSF, in 1986, sponsored NSFNET, a backbone that connected five supercomputer centers located throughout the United States. Community networks were allowed access to this backbone, a T-1 line with a 1.544-Mbps data rate, thus providing connectivity throughout the United States. In 1990, ARPANET was officially retired and replaced by NSFNET. In 1995, NSFNET reverted back to its original concept of a research network.

ANSNET

In 1991, the U.S. government decided that NSFNET was not capable of supporting the rapidly increasing Internet traffic. Three companies, IBM, Merit, and MCI, filled the void by forming a nonprofit organization called Advanced Network and Services (ANS) to build a new, high-speed Internet backbone called ANSNET.

The Internet Today

The Internet today is not a simple hierarchical structure. It is made up of many wide and local area networks joined by connecting devices and switching stations. It is difficult to give an accurate representation of the Internet because it is continuously changing—new networks are being added, existing networks need more addresses, and networks of defunct companies need to be removed. Today most end users who want Internet connection use the services of Internet service providers (ISPs). There are international service providers, national service providers, regional service providers, and local service providers. The Internet today is run by private companies, not the government. Figure shows a conceptual (not geographical) view of the Internet.

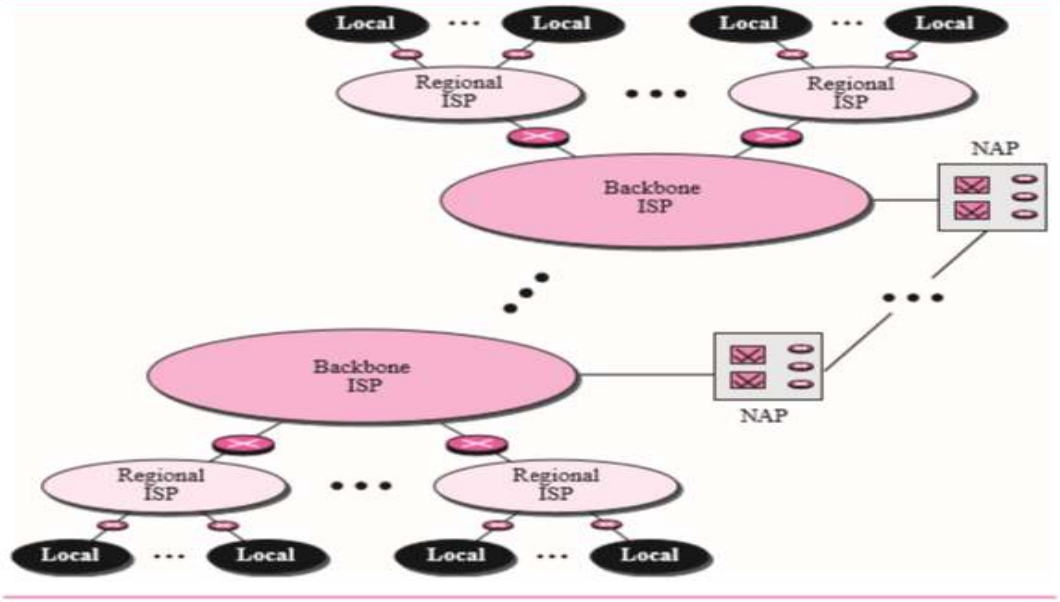
Backbone ISPs:

Backbone ISPs are created and maintained by specialized companies. There are many backbone ISPs operating in North America; some of the most well-known are SprintLink, PSINet, UUNet Technology, AGIS, and internet MCI. To provide connectivity between the end users, these backbone networks are connected by complex switching stations (normally run by a third party) called network access points (NAPs). Some regional ISP networks are also connected to each other by private switching stations called peering points. Backbone ISPs normally operate at a high data rate (10 Gbps, for example).

Regional ISPs:

Regional ISPs are small ISPs that are connected to one or more backbone ISPs. They are at the second level of hierarchy with a lesser data rate.

Figure *Internet today*



Local ISPs:

Local ISPs provide direct service to the end users. The local ISPs can be connected to regional ISPs or directly to backbone ISPs. Most end users are connected to the local ISPs. Note that in this sense, a local ISP can be a company that just provides Internet services, a corporation with a network to supply services to its own employees, or a nonprofit organization, such as a college or a university, that runs its own network. Each of these can be connected to a regional or backbone service provider.

World Wide Web

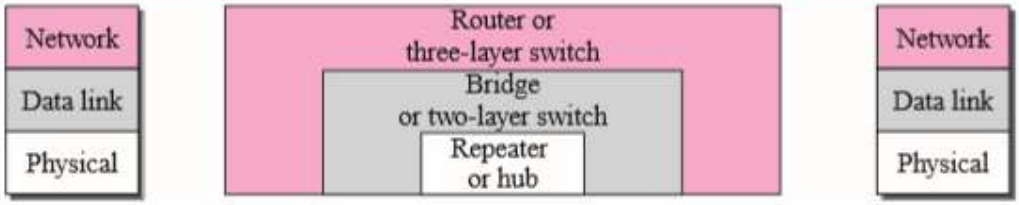
The 1990s saw the explosion of the Internet applications due to the emergence of the World Wide Web (WWW). The web was invented at CERN by Tim Berners-Lee. This invention has added the commercial applications to the Internet.

2. Interconnecting Devices

LANs or WANs do not normally operate in isolation. They are connected to one another or to the Internet. To connect LANs and WANs together we use connecting devices. Connecting devices can operate in different layers of the Internet model.

We discuss three kinds of connecting devices: repeaters (or hubs), bridges (or two-layer switches), and routers (or three-layer switches). Repeaters and hubs operate in the first layer of the Internet model. Bridges and two-layer switches operate in the first two layers. Routers and three-layer switches operate in the first three layers. Figure shows the layers in which each device operates.

Figure *Connecting devices*

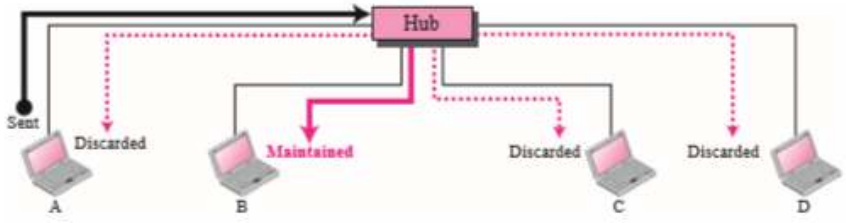


i) Repeaters

A repeater is a device that operates only in the physical layer. Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data. A repeater receives a signal and, before it becomes too weak or corrupted, regenerates and retimes the original bit pattern. The repeater then sends the refreshed signal. In the past, when Ethernet LANs were using bus topology, a repeater was used to connect two segments of a LAN to overcome the length restriction of the coaxial cable. Today, however, Ethernet LANs use star topology. In a star topology, a repeater is a multiport device, often called a hub that can be used to serve as the connecting point and at the same time function as a repeater.

The following figure shows that when a packet from station A to B arrives at the hub, the signal representing the frame is regenerated to remove any possible corrupting noise, but the hub forwards the packet from all outgoing port to all stations in the LAN. In other words, the frame is broadcast. All stations in the LAN receive the frame, but only station B keeps it. The rest of the stations discard it. Figure also shows the role of a repeater or a hub in a switched LAN. The figure definitely shows that a hub does not have a filtering capability; it does not have the intelligence to find from which port the frame should be sent out.

Figure 3.41 *Repeater or hub*



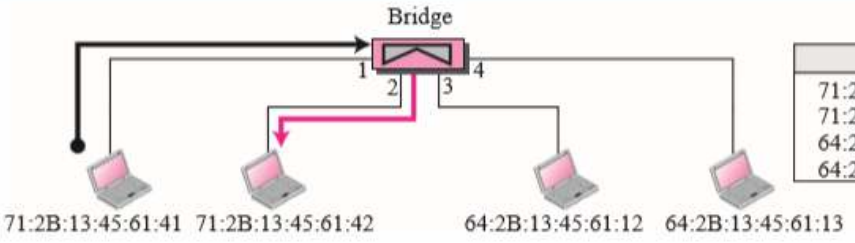
A hub or a repeater is a physical-layer device. They do not have any data-link address and they do not check the data-link address of the received frame. They just regenerate the corrupted bits and send them out from every port.

ii) Bridges

A bridge operates in both the physical and the data link layers. As a physical-layer device, it regenerates the signal it receives. As a data link layer device, the bridge can check the MAC addresses (source and destination) contained in the frame.

Filtering: A bridge has filtering capability. It can check the destination address of a frame and can decide from which outgoing port the frame should be send out. Let us give an example. The following figure, we have a LAN with four stations that are connected to a bridge. If a frame destined for station 71:2B:13:45:61:42 arrives at port 1, the bridge consults its table to find the departing port. According to its table, frames for 71:2B:13:45:61:42 should be sent out only through port 2; therefore, there is no need for forwarding the frame through other ports

Figure *Bridge*



Bridge table

Address	Port
71:2B:13:45:61:41	1
71:2B:13:45:61:42	2
64:2B:13:45:61:12	3
64:2B:13:45:61:13	4

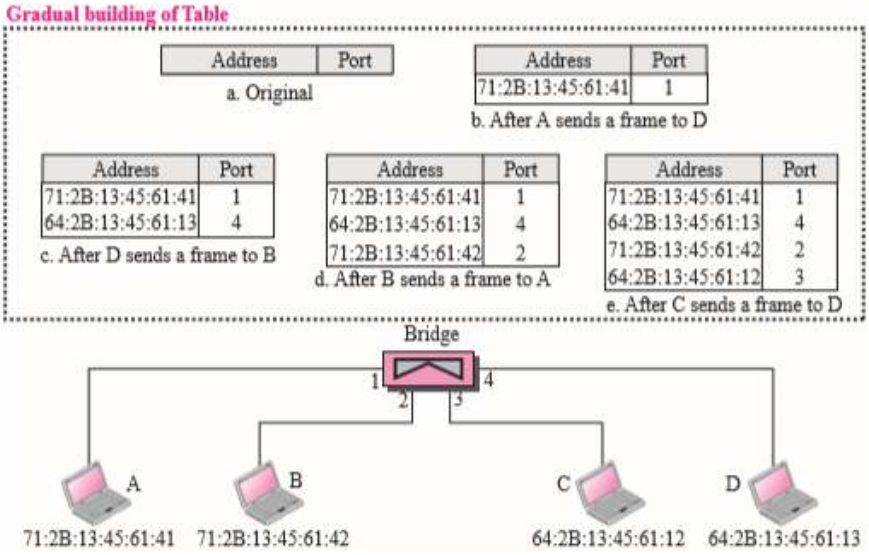
Transparent Bridges: A transparent bridge is a bridge in which the stations are completely unaware of the bridge's existence. If a bridge is added or deleted from the system, reconfiguration of the stations is unnecessary. According to the IEEE 802.1d specification, a system equipped with transparent bridges must meet three criteria: 1. Frames must be forwarded from one station to another. 2. The forwarding table is automatically made by learning frame movements in the network. 3. Loops in the system must be prevented.

Forwarding: A transparent bridge must correctly forward the frames, as discussed in the previous section.

Learning: The earliest bridges had forwarding tables that were static. The system administrator would manually enter each table entry during bridge setup. Although the process was simple, it was not practical. If a station was added or deleted, the table had to be modified manually. The same was true if a

station's MAC address changed, which is not a rare event. For example, putting in a new network card means a new MAC address. A better solution to the static table is a dynamic table that maps addresses to ports automatically. To make a table dynamic, we need a bridge that gradually learns from the frame movements. To do this, the bridge inspects both the destination and the source addresses. The destination address is used for the forwarding decision (table lookup); the source address is used for adding entries to the table and for updating purposes. Let us elaborate on this process using Figure

Figure *Learning bridge*



1. When station A sends a frame to station D, the bridge does not have an entry for either D or A. The frame goes out from all three ports; the frame floods the network. However, by looking at the source address, the bridge learns that station A must be connected to port 1. This means that frames destined for A, in the future, must be sent out through port 1. The bridge adds this entry to its table. The table has its first entry now.
2. When station D sends a frame to station B, the bridge has no entry for B, so it floods the network again. However, it adds one more entry to the table.
3. The learning process continues until the table has information about every port.

Two-Layer Switch: When we use the term switch, we must be careful because a switch can mean two different things. We must clarify the term by adding the level at which the device operates. We can have a two-layer switch or a three-layer switch. A two-layer switch performs at the physical and data link layer; it is a sophisticated bridge with faster forwarding capability.

iii) Routers

A router is a three-layer device; it operates in the physical, data link, and network layers. As a physical layer device, it regenerates the signal it receives. As a data link layer device, the router checks the physical addresses (source and destination) contained in the packet. As a network layer device, a router checks the network layer addresses (addresses in the IP layer). Note that bridges change collision domains, but routers limit broadcast domains.

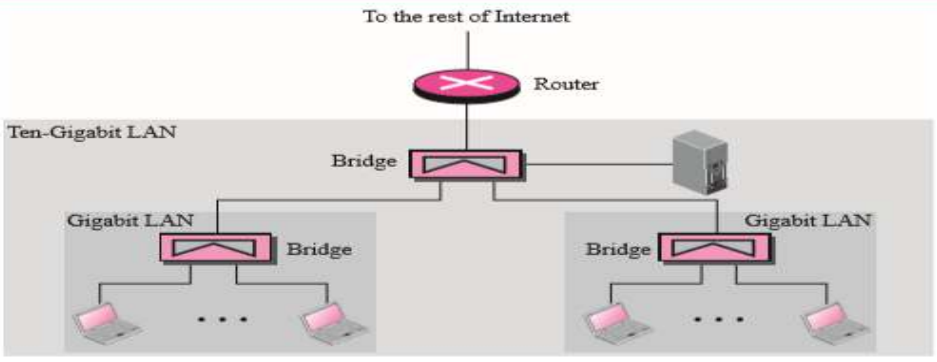
A router can connect LANs together; a router can connect WANs together; and a router can connect LANs and WANs together. In other words, a router is an internetworking device; it connects independent networks together to form an internetwork. According to this definition, two networks (LANs or WANs) connected by a router become an internetwork or an internet.

There are three major differences between a router and a repeater or a bridge.

1. A router has a physical and logical (IP) address for each of its interfaces.
2. A router acts only on those packets in which the physical destination address matches the address of the interface at which the packet arrives.
3. A router changes the physical address of the packet (both source and destination) when it forwards the packet.

Let us give an example. In the following Figure we assume an organization has two separate buildings with a Gigabit Ethernet LANs installed in each building. The organization uses bridges in each LAN. The two LANs can be connected together to form a larger LAN using Ten-Gigabit Ethernet technology that speeds up the connection to the Ethernet and the connection to the organization server. A router then can connect the whole system to the Internet. A router will change the MAC address it receives because the MAC addresses have only local jurisdictions.

Figure 3.44 *Routing example*



Three-Layer Switch: A three-layer switch is a router; a router with an improved design to allow better performance. A three-layer switch can receive, process, and dispatch a packet much faster than a traditional router even though the functionality is the same.

3. Protocol and Standards

This Section define two widely used terms: protocols and standards. First, we define protocol, which is synonymous with "rule." Then we discuss standards, which are agreed-upon rules.

Protocols

Communication between two people or two devices needs to follow some protocol. A protocol is a set of rules that governs communication. In a telephone conversation, there are a set of rules that we need to follow. There is a rule how to make connection (dialing the telephone number), how to respond to the call (picking up the receiver), how to greet, how to let the communication flow smoothly by listening when the other party is talking, and finally how to end the communication (hanging up).

In computer networks, communication occurs between entities in different systems. An entity is anything capable of sending or receiving information. However, two entities cannot simply send bit streams to each other and expect to be understood. For communication to occur, the entities must agree on a protocol. A protocol defines what is communicated, how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics, and timing.

Syntax: Syntax refers to the structure or format of the data, meaning the order in which they are presented. For example, a simple protocol might expect the first 8 bits of data to be the address of the sender, the second 8 bits to be the address of the receiver, and the rest of the stream to be the message itself. The data order is also applied to the order of bits when they are stored or transmitted. Different computers may store data in different bit orders. When these computers communicate, this difference needs to be resolved.

Semantics: Semantics refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and what action is to be taken based on that interpretation? For example, does an address identify the route to be taken or the final destination of the message?

Timing: Timing refers to two characteristics: when data should be sent and how fast it can be sent. For example, if a sender produces data at 100 megabits per second (100 Mbps) but the receiver can process data at only 1 Mbps, the transmission will overload the receiver and data will be largely lost.

Standards

Standards are essential in creating and maintaining an open and competitive market for equipment manufacturers and also in guaranteeing national and international interoperability of data and

telecommunications technology and processes. They provide guidelines to manufacturers, vendors, government agencies, and other service providers to ensure the kind of interconnectivity necessary in today's marketplace and in international communications.

Data communication standards fall into two categories: de facto (meaning "by fact" or "by convention") and de jure (meaning "by law" or "by regulation").

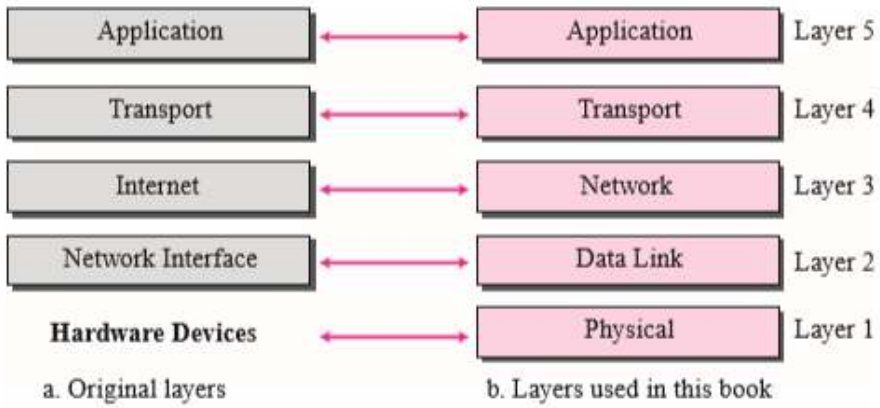
De facto: Standards that have not been approved by an organized body but have been adopted as standards through widespread use are de facto standards. De facto standards are often established originally by manufacturers that seek to define the functionality of a new product or technology. Examples of de facto standards are MS Office and various DVD standards.

De jure: De jure standards are those that have been legislated by an officially recognized body.

4. TCP/IP Protocol Suite

The TCP/IP protocol suite was developed prior to the OSI model. The original TCP/IP protocol suite was defined as four software layers built upon the hardware. Today, however, TCP/IP is thought of as a five-layer model with the layers named similarly to the ones in the OSI model.

Figure *Layers in the TCP/IP Protocol Suite*

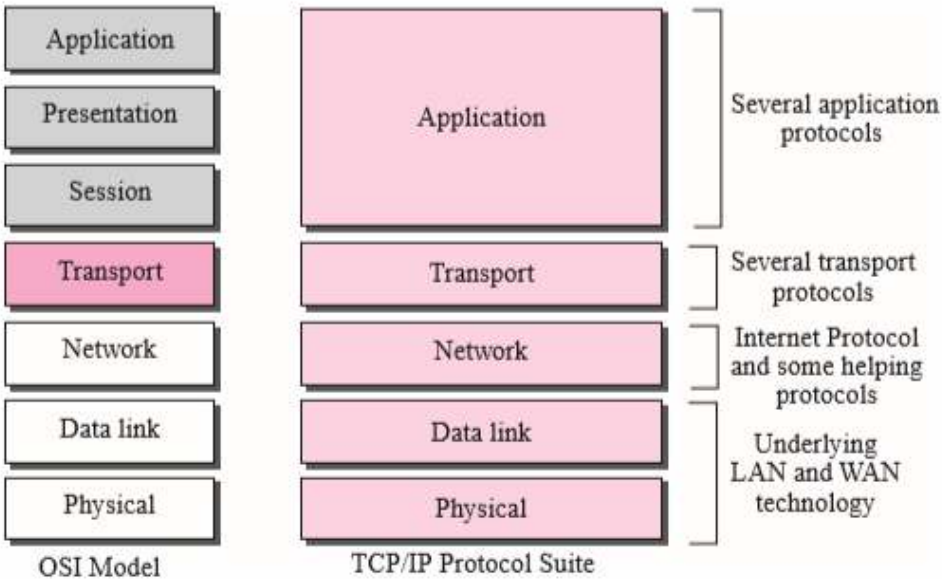


Comparison between OSI and TCP/IP Protocol Suite

Two reasons were mentioned for this decision. First, TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport layer protocols. Second, the application layer is not only one piece of software. Many applications can be developed at this layer. If some of the functionalities mentioned in the session and presentation are needed for a particular application, it can be included in the development of that piece of software. TCP/IP is a

hierarchical protocol made up of interactive modules, each of which provides a specific functionality, but the modules are not necessarily interdependent. Whereas the OSI model specifies which functions belong to each of its layers, the layers of the TCP/IP protocol suite contain relatively independent protocols that can be mixed and matched, depending on the needs of the system.

Figure *TCP/IP and OSI model*



Physical Link Layer

TCP/IP does not define any specific protocol for the physical layer. It supports all of the standard and proprietary protocols. At this level, the communication is between two hops or nodes, either a computer or router. The unit of communication is a single bit. When the connection is established between the two nodes, a stream of bits is flowing between them. The physical layer, however, treats each bit individually. We are assuming that at this moment the two computers have discovered that the most efficient way to communicate with each other is via routers.

Note that if a node is connected to n links, it needs n physical-layer protocols, one for each link. The reason is that different links may use different physical-layer protocols. Each computer involves with only one link; each router involves with only two links.

The responsibility of the physical layer, in addition to delivery of bits, matches with what mentioned for the physical layer of the OSI model, but it mostly depends on the underlying technologies that provide links.

Data Link Layer

TCP/IP does not define any specific protocol for the data link layer either. It supports all of the standard and proprietary protocols. At this level, the communication is also between two hops or nodes. The unit of communication however, is a packet called a frame. A frame is a packet that encapsulates the data received from the network layer with an added header and sometimes a trailer. The head, among other communication information, includes the source and destination of frame. The destination address is needed to define the right recipient of the frame because many nodes may have been connected to the link. The source address is needed for possible response or acknowledgment as may be required by some protocols.

Network Layer

At the network layer (or, more accurately, the internetwork layer), TCP/IP supports the Internet Protocol (IP). The Internet Protocol (IP) is the transmission mechanism used by the TCP/IP protocols. IP transports data in packets called datagrams, each of which is transported separately. Datagrams can travel along different routes and can arrive out of sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination.

Note that there is a main difference between the communication at the network layer and the communication at data link or physical layers. Communication at the network layer is end to end while the communications at the other two layers are node to node.

Transport Layer

There is a main difference between the transport layer and the network layer. Although all nodes in a network need to have the network layer, only the two end computers need to have the transport layer. The network layer is responsible for sending individual datagrams from computer A to computer B; the transport layer is responsible for delivering the whole message, which is called a segment, a user datagram, or a packet, from A to B. A segment may consist of a few or tens of datagrams. The segments need to be broken into datagrams and each datagram has to be delivered to the network layer for transmission. Since the Internet defines a different route for each datagram, the datagrams may arrive out of order and may be lost. The transport layer at computer B needs to wait until all of these datagrams to arrive, assemble them and make a segment out of them.

Again, we should know that the two transport layers only think that they are communicating with each other using a segment; the communication is done through the physical layer and the exchange of bits. Traditionally, the transport layer was represented in the TCP/IP suite by two protocols: User

Datagram Protocol (UDP) and Transmission Control Protocol (TCP). A new protocol called Stream Control Transmission Protocol (SCTP) has been introduced in the last few years.

Application Layer

The application layer in TCP/IP is equivalent to the combined session, presentation, and application layers in the OSI model. The application layer allows a user to access the services of our private internet or the global Internet. Many protocols are defined at this layer to provide services such as electronic mail, file transfer, accessing the World Wide Web, and so on.

Note that the communication at the application layer, like the one at the transport layer, is end to end. A message generated at computer A is sent to computer B without being changed during the transmission.

5. Routing Protocol

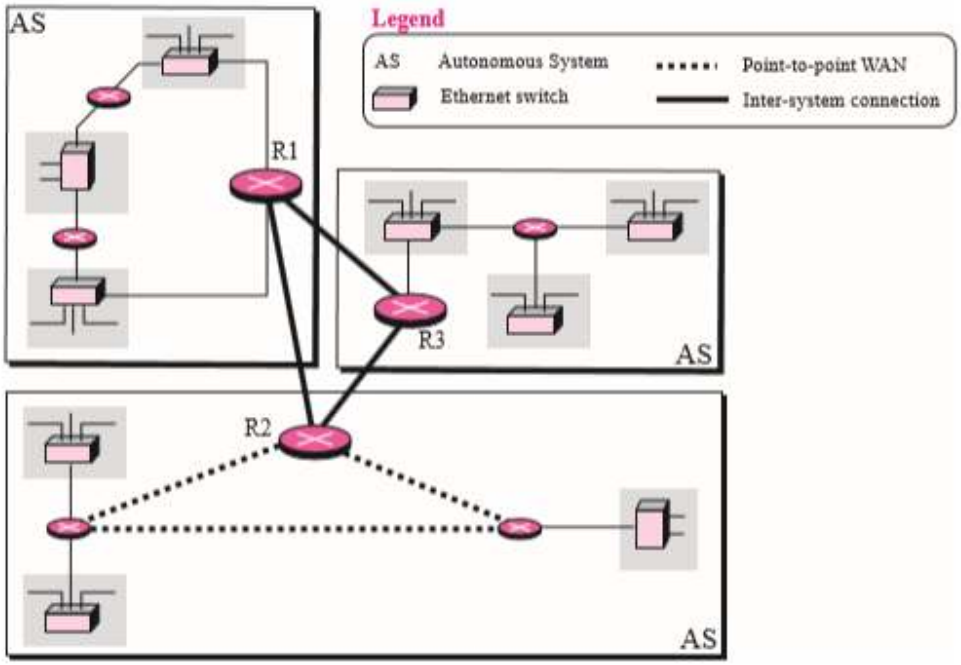
Routing protocols have been created in response to the demand for dynamic routing tables. A routing protocol is a combination of rules and procedures that let routers in the internet inform each other of changes. It allows routers to share whatever they know about the internet or their neighborhood. The sharing of information allows a router in San Francisco to know about the failure of a network in Texas. The routing protocols also include procedures for combining information received from other routers. Routing protocols can be either an interior protocol or an exterior protocol. An interior protocol handles intra-domain routing; an exterior protocol handles inter-domain routing.

Intra and Inter Domain Routing: Internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems.

An autonomous system (AS) is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as intra-domain routing. Routing between autonomous systems is referred to as inter-domain routing. Each autonomous system can choose one or more intra-domain routing protocols to handle routing inside the autonomous system. However, only one inter-domain routing protocol handles routing between autonomous systems.

Routing Information Protocol (RIP) is the implementation of the distance vector protocol. Open Shortest Path First (OSPF) is the implementation of the link state protocol. Border Gateway Protocol (BGP) is the implementation of the path vector protocol. RIP and OSPF are interior routing protocols; BGP is an exterior routing protocol.

Figure *Autonomous systems*

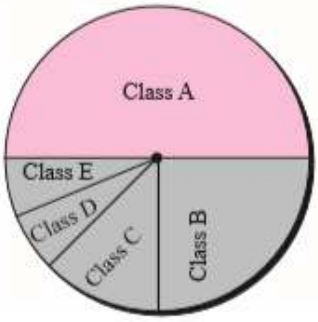


6. Classful IP Addressing

IP addresses, when started a few decades ago, used the concept of classes. This architecture is called classful addressing.

Classes: In classful addressing, the IP address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the whole address space. The following Figure shows the class occupation of the address space.

Figure *Occupation of the address space*



Class A: 2^{31} = 2,147,483,648 addresses, 50%

Class B: 2^{30} = 1,073,741,824 addresses, 25%

Class C: 2^{29} = 536,870,912 addresses, 12.5%

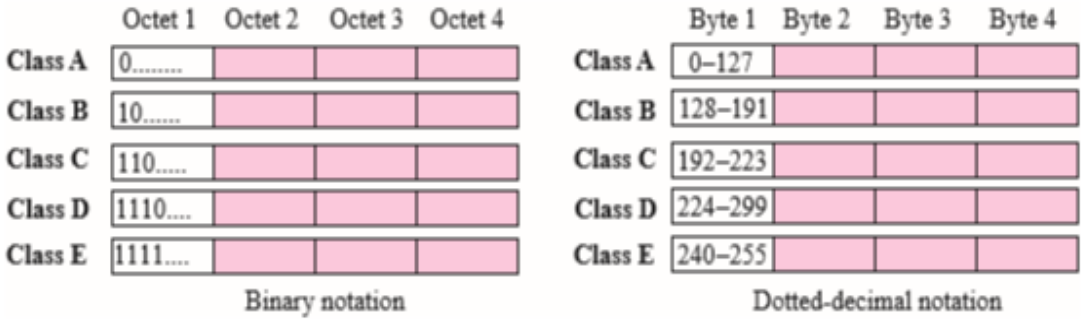
Class D: 2^{28} = 268,435,456 addresses, 6.25%

Class E: 2^{28} = 268,435,456 addresses, 6.25%

Recognizing Classes

We can find the class of an address when the address is given either in binary or in dotted decimal notation. In the binary notation, the first few bits can immediately tell us the class of the address; in the dotted-decimal notation, the value of the first byte can give the class of an address.

Figure *Finding the class of an address*



Note that some special addresses fall in class A or E. We emphasize that these special addresses are exceptions to the classification;

Ex:

Find the class of each address:

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 10100111 11011011 10001011 01101111

d. 11110011 10011011 11111011 00001111

Answer

a. The first bit is 0. This is a class A address.

b. The first 2 bits are 1; the third bit is 0. This is a class C address.

c. The first bit is 1; the second bit is 0. This is a class B address.

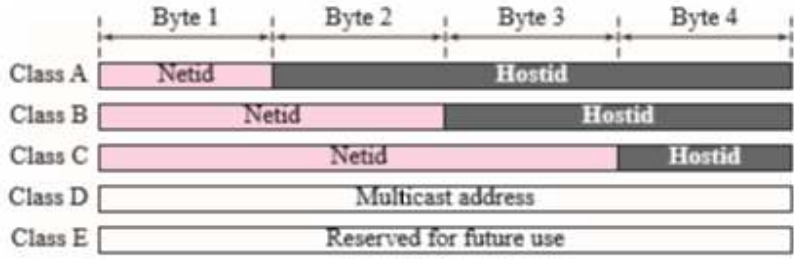
d. The first 4 bits are 1s. This is a class E address.

Netid and Hostid

In classful addressing, an IP address in classes A, B, and C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address. The following figure shows the netid and hostid bytes. Note that classes D and E are not divided into netid and hosted. In class A, 1 byte

defines the netid and 3 bytes define the hostid. In class B, 2 bytes define the netid and 2 bytes define the hostid. In class C, 3 bytes define the netid and 1 byte defines the hostid.

Figure *Netid and hostid*



Classes and Blocks: One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size. Let us look at each class.

Class A: Since only 1 byte in class A defines the netid and the leftmost bit should be 0, the next 7 bits can be changed to find the number of blocks in this class. Therefore, class A is divided into 27 = 128 blocks that can be assigned to 128 organizations (the number is less because some blocks were reserved as special blocks). However, each block in this class contains 16,777,216 addresses, which means the organization should be a really large one to use all these addresses. Many addresses are wasted in this class. Figure 5.9 shows the block in class A.

Class B: Since 2 bytes in class B define the class and the two leftmost bit should be 10 (fixed), the next 14 bits can be changed to find the number of blocks in this class. Therefore, class B is divided into 214 = 16,384 blocks that can be assigned to 16,384 organizations (the number is less because some blocks were reserved as special blocks). However, each block in this class contains 65,536 addresses. Not so many organizations can use so many addresses. Many addresses are wasted in this class. Figure 5.10 shows the blocks in class B.

Class C: Since 3 bytes in class C define the class and the three leftmost bits should be 110 (fixed), the next 21 bits can be changed to find the number of blocks in this class. Therefore, class C is divided into 221 = 2,097,152 blocks, in which each block contains 256 addresses that can be assigned to 2,097,152 organizations (the number is less because some blocks were reserved as special blocks). Each block contains 256 addresses. However, not so many organizations were so small as to be satisfied with a class C block. Figure 5.11 shows the blocks in class C.

Class D: There is just one block of class D addresses. It is designed for multicasting, as we will see in a later section. Each address in this class is used to define one group of hosts on the Internet. When a group

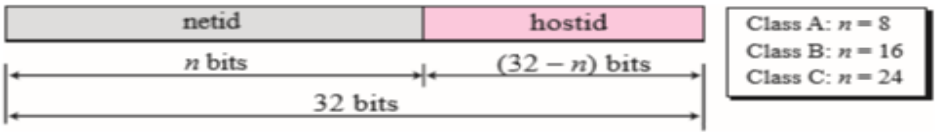
is assigned an address in this class, every host that is a member of this group will have a multicast address in addition to its normal (unicast) address.

Class E: There is just one block of class E addresses. It was designed for use as reserved addresses.

Two-Level Addressing

When classful addressing was designed, it was assumed that the whole Internet is divided into many networks and each network connects many hosts. In other words, the Internet was seen as a network of networks. A network was normally created by an organization that wanted to be connected to the Internet. The Internet authorities allocated a block of addresses to the organization (in class A, B, or C). Since all addresses in a network belonged to a single block, each address in classful addressing contains two parts: netid and hostid. The netid defines the network; the hostid defines a particular host connected to that network. If n bits in the class defines the net, then 32 − n bits defines the host. However, the value of n depends on the class the block belongs to. The value of n can be 8, 16 or 24 corresponding to classes A, B, and C respectively.

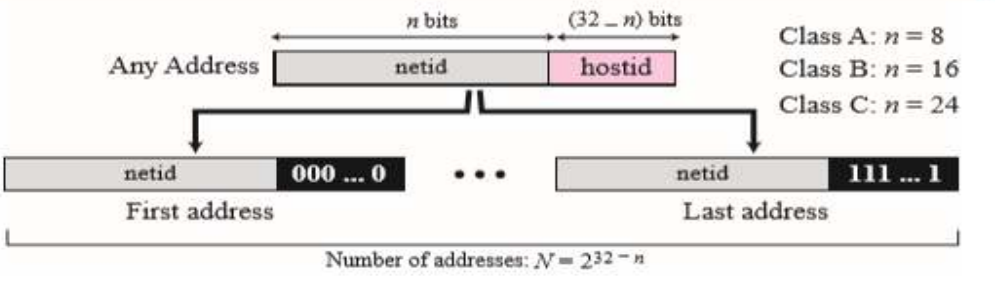
Figure *Two-level addressing in classful addressing*



Extracting Information in a Block:

A block is a range of addresses. Given any address in the block, we normally like to know three pieces of information about the block: the number of addresses, the first address, and the last address. Before we can extract these pieces of information, we need to know the class of the address, which we showed how to find in the previous section. After the class of the block is found, we know the value of n, the length of netid in bits.

Figure *Information extraction in classful addressing*



We can now find these three pieces of information in the above figure.

1. The number of addresses in the block, N, can be found using N = 232−n.
2. To find the first address, we keep the n leftmost bits and set the (32 − n) rightmost bits all to 0s.
3. To find the last address, we keep the n leftmost bits and set the (32 − n) rightmost bits all to 1s.

Example

An address in a block is given as 73.22.17.25. Find the number of addresses in the block, the first address, and the last address.

Solution

Since 73 is between 0 and 127, the class of the address is A. The value of n is 8.

1. The number of addresses in this block is N = 232−n = 224 = 16,777,216.
2. To find the first address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 0s. The first address is 73.0.0.0/8 in which 8 is the value of n. The first address is called the network address and is not assigned to any host. It is used to define the network.
3. To find the last address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 1s. The last address is 73.255.255.255. The last address is normally used for a special purpose.

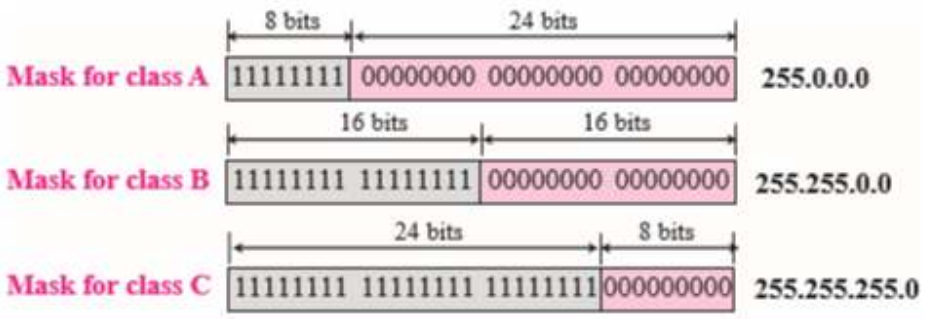
Network Address

The above three examples show that, given any address, we can find all information about the block. The first address (network address), is particularly important because it is used in routing a packet to its destination network. For the moment, let us assume that an internet is made of m networks and a router with m interfaces. When a packet arrives at the router from any source host, the router needs to know to which network the packet should be sent; the router needs to know from which interface the packet should be sent out.

Network Mask

The routers in the Internet normally use an algorithm to extract the network address from the destination address of a packet. To do this, we need a network mask. A network mask or a default mask in classful addressing is a 32-bit number with n leftmost bits all set to 1s and (32 − n) rightmost bits all set to 0s. Since n is different for each class in classful addressing, we have three default masks in classful addressing as shown in Figure. To extract the network address from the destination address of a packet, a router uses the AND operation. When the destination address (or any address in the block) is ANDed with the default mask, the result is the network address.

Figure *Network mask*



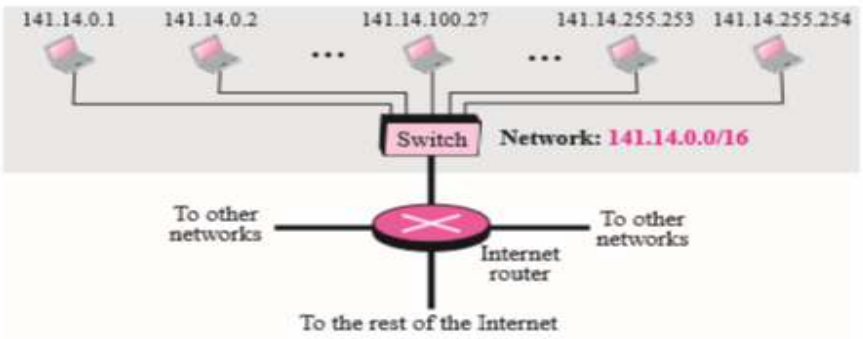
Three-Level Addressing: Subnetting

To reach a host on the Internet, we must first reach the network and then the host. It soon became clear that we need more than two hierarchical levels, for two reasons. First, an organization that was granted a block in class A or B needed to divide its large network into several subnetworks for better security and management. Second, since the blocks in class A and B were almost depleted and the blocks in class C were smaller than the needs of most organizations, an organization that has been granted a block in class A or B could divide the block into smaller subblocks and share them with other organizations. The idea of splitting a block to smaller blocks is referred to as subnetting. In subnetting, a network is divided into several smaller subnetworks (subnets) with each subnetwork having its own subnetwork address.

Example:

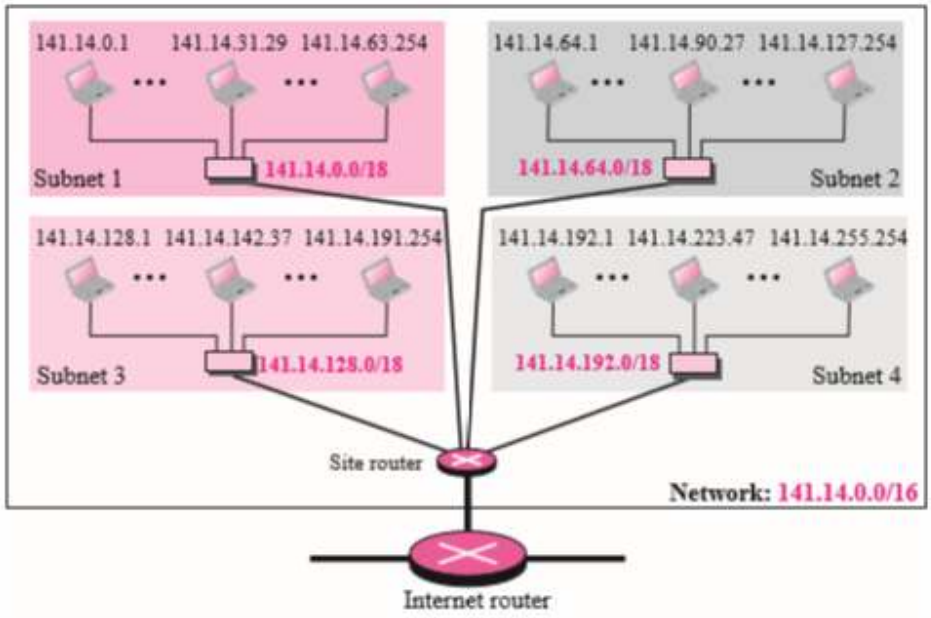
The Following figure shows a network using class B addresses before subnetting. We have just one network with almost 216 hosts. The whole network is connected, through one single connection, to one of the routers in the Internet. Note that we have shown /16 to show the length of the netid (class B).

Figure



The following figure show after subnetting. The whole network is still connected to the Internet through the same router. However, the network has used a private router to divide the network into four subnetworks. The rest of the Internet still sees only one network; internally the network is made of four subnetworks. Each subnetwork can now have almost 214 hosts.

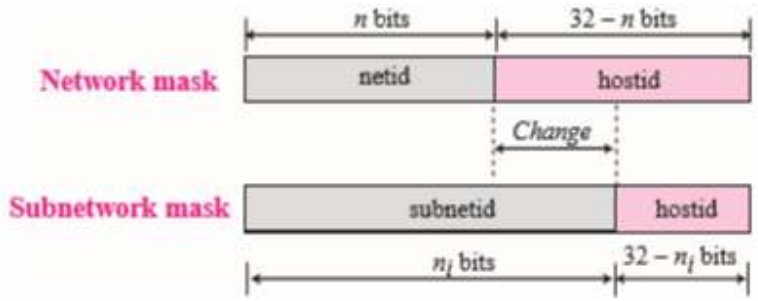
Figure



Subnet Mask

When we divide a network to several subnetworks, we need to create a subnetwork mask (or subnet mask) for each subnetwork. A subnetwork has subnetid and hostid as shown in the following figure.

Figure *Network mask and subnetwork mask*



Subnetting increases the length of the netid and decreases the length of hostid. When we divide a network to s number of subnetworks, each of equal numbers of hosts, we can calculate the subnetid for each subnetwork as

$$nsub = n + log2s$$

in which n is the length of netid, nsub is the length of each subnetid, and s is the number of subnets which must be a power of 2.

Supernetting

In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork. By doing this, an organization can apply for several class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four class C blocks.

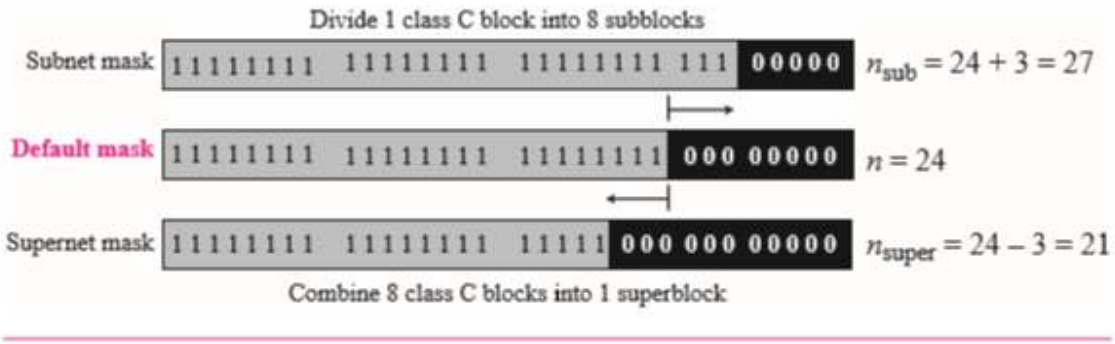
Supernet Mask

A supernet mask is the reverse of a subnet mask. A subnet mask for class C has more 1s than the default mask for this class. A supernet mask for class C has less 1s than the default mask for this class.

Subnet vs Supernet

The following figure shows the difference between a subnet mask and a supernet mask. A subnet mask that divides a block into eight subblocks has three more 1s (23 = 8) than the default mask; a supernet mask that combines eight blocks into one superblock has three less 1s than the default mask.

Figure *Comparison of subnet, default, and supernet masks*



In supernetting, the number of class C addresses that can be combined to make a supernet needs to be a power of 2. The length of the supernetid can be found using the formula

$$nsuper = n - log2c$$

in which nsuper defines the length of the supernetid in bits and c defines the number of class C blocks that are combined.

Unfortunately, supernetting provided two new problems:

1. The number of blocks to combine needs to be a power of 2, which means an organization that needed seven blocks should be granted at least eight blocks (address wasting).
2. Supernetting and subnetting really complicated the routing of packets in the Internet.

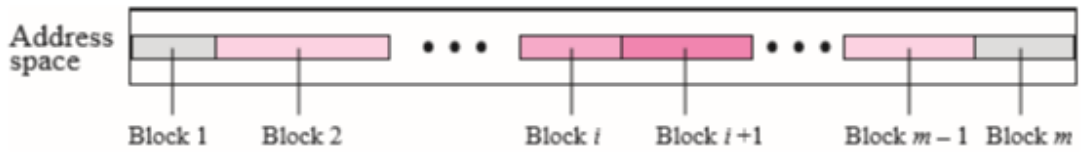
7. Classless Addressing

Subnetting and supernetting in classful addressing did not really solve the address depletion problem and made the distribution of addresses and the routing process more difficult. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses to be increased, which means the format of the IP packets needs to be changed. The short-term solution still used in IPv4 addresses is called classless addressing. In other words, the class privilege was removed from the distribution to compensate for the address depletion.

Variable-Length Blocks

In classless addressing, the whole address space is divided into variable length blocks. Theoretically, we can have a block of 2^0, 2^1, 2^2. . . 2^{32} addresses. The only restriction is that the number of addresses in a block needs to be a power of 2. An organization can be granted one block of addresses. The following figure shows the division of the whole address space into non-overlapping blocks.

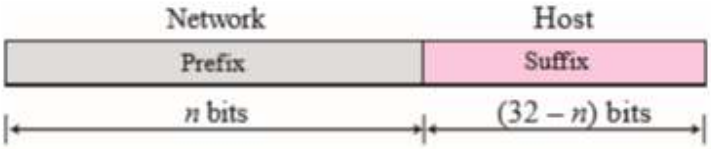
Figure *Variable-length blocks in classless addressing*



Two-Level Addressing

In classful addressing, two-level addressing was provided by dividing an address into netid and hostid. The same idea can be applied in classless addressing. When an organization is granted a block of addresses, the block is actually divided into two parts, the prefix and the suffix. The prefix plays the same role as the netid; the suffix plays the same role as the hostid. All addresses in the block have the same prefix; each address has a different suffix. The following figure shows the prefix and suffix in a classless block.

Figure *Prefix and suffix*



In classful addressing, the length of the netid, n, depends on the class of the address; it can be only 8, 16, or 24. In classless addressing, the length of the prefix, n, depends on the size of the block; it can be 0, 1, 2, 3, . . . , 32. In classless addressing, the value of n is referred to as prefix length; the value of 32 − n is referred to as suffix length.

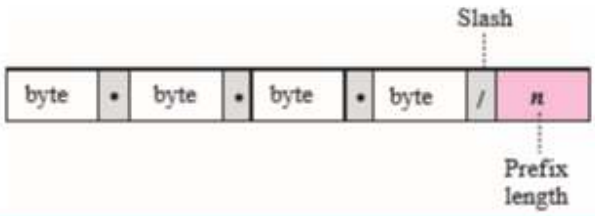
Slash Notation

The netid length in classful addressing or the prefix length in classless addressing play a very important role when we need to extract the information about the block from a given address in the block. However, there is a difference here in classful and classless addressing.

1. In classful addressing, the netid length is inherent in the address. Given an address, we know the class of the address that allows us to find the netid length (8, 16, or 24).
2. In classless addressing, the prefix length cannot be found if we are given only an address in the block. The given address can belong to a block with any prefix length. In classless addressing, we need to include the prefix length to each address if we need to find the block of the address. In this case, the prefix length, n, is added to the address separated by a slash. The notation is informally referred to as slash notation.

An address in classless addressing can then be represented as shown below

Figure *Slash notation*



The slash notation is formally referred to as classless interdomain routing or CIDR (pronounced cider) notation.

Extracting Block Information

An address in slash notation (CIDR) contains all information we need about the block: the first address (network address), the number of addresses, and the last address. These three pieces of information can be found as follows:

- The number of addresses in the block can be found as:

  $$N = 232 - n$$

  in which n is the prefix length and N is the number of addresses in the block.
- The first address (network address) in the block can be found by ANDing the address with the network mask:

  First address = (any address) AND (network mask)

  Alternatively, we can keep the n leftmost bits of any address in the block and set the 32− n bits to 0s to find the first address.
- The last address in the block can be found by either adding the first address with the number of addresses or, directly, by ORing the address with the complement (NOTing) of the network mask:

  Last address = (any address) OR [NOT (network mask)]

  Alternatively, we can keep the n leftmost bits of any address in the block and set the 32 − n bits to 1s to find the last address.

Block Allocation

The next issue in classless addressing is block allocation. How are the blocks allocated? The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Addresses (ICANN). However, ICANN does not normally allocate addresses to individual Internet users. It assigns a large block of addresses to an ISP (or a larger organization that is considered an ISP in this case). For the proper operation of the CIDR, three restrictions need to be applied to the allocated block. 1. The number of requested addresses, N, needs to be a power of 2. This is needed to provide an integer value for the prefix length, n (see the second restriction). The number of addresses can be 1, 2, 4, 8, 16, and so on. 2. The value of prefix length can be found from the number of addresses in the block. Since N = 232 − n, then n = log2 (232/N) = 32 − log2N. That is the reason why N needs to be a power of 2. 3. The requested block needs to be allocated where there are a contiguous number of unallocated addresses in the address space. However, there is a restriction on choosing the beginning addresses of the block. The beginning address needs to be divisible by the number of addresses in the block. To see this restriction, we can show that the beginning address can be calculated as X × 2n − 32 in which X is the decimal value of the prefix. In other words, the beginning address is X × N.

Subnetting

Three levels of hierarchy can be created using subnetting. An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a subnetwork (or subnet). The concept is the same as we discussed for classful addressing. Note that nothing stops the organization from creating more levels. A subnetwork can be divided into several sub-subnetworks. A sub-subnetwork can be divided into several sub-sub-subnetworks. And so on.

Designing Subnets

The subnetworks in a network should be carefully designed to enable the routing of packets. We assume the total number of addresses granted to the organization is N, the prefix length is n, the assigned number of addresses to each subnetwork is Nsub, the prefix length for each subnetwork is nsub, and the total number of subnetworks is s. Then, the following steps need to be carefully followed to guarantee the proper operation of the subnetworks.

1. The number of addresses in each subnetwork should be a power of 2.
2. The prefix length for each subnetwork should be found using the following formula:

$$nsub = n + \log2\ (N/Nsub)$$

3. The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork. This can be achieved if we first assign addresses to larger networks.

Address Aggregation

One of the advantages of CIDR architecture is address aggregation. ICANN assigns a large block of addresses to an ISP. Each ISP in turn divides its assigned block into smaller subblocks and grants the subblocks to its customers; many blocks of addresses are aggregated in one block and granted to one ISP.

Possible Questions

Two Mark Questions

1. Define a network?
2. List the layer of OSI.
3. What is an IP Address?
4. What is the drawback of Classful Addressing?
5. Define subnetting.

Six Mark Questions

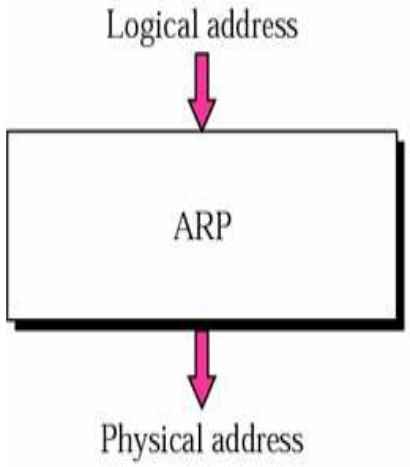
1. Write a detail note on TCP/IP protocol suite.
2. What is topology? Discuss the different topology available in network.
3. Discuss in detail about the difference between TCP/IP Protocol suite and OSI Reference Model.
4. List the set of connecting devices. Explain about it in detail.
5. Write a detail note on Classful addressing.
6. Discuss about Sub-netting along classful addressing.
7. Find the network address, last address, number of machines, ranges of machines for the IP 32.16.8.4/8
8. Find the network address, last address, number of machines, ranges of machines for the IP 32.16.8.4
9. Discuss in detail about Classless Addressing.
10. Write a detail note on OSI Reference model.

SYLLABUS

ARP & RARP – Proxy ARP – ARP over ATM – ARP and RAPP protocol format. IP datagram – Fragmentation – Options – IP datagram format – Routing IP datagrams – Cheksum. ICMP – types of messages – message format – Error reporting – Query – Cheksum

1. ARP (Address Resolution Protocol)

The address resolution protocol is a protocol used by the Internet Protocol (IP) [RFC826], specifically IPv4, to map IP network addresses to the hardware addresses used by a data link protocol. The protocol operates below the network layer as a part of the interface between the OSI network and OSI link layer. It is used when IPv4 is used over Ethernet.

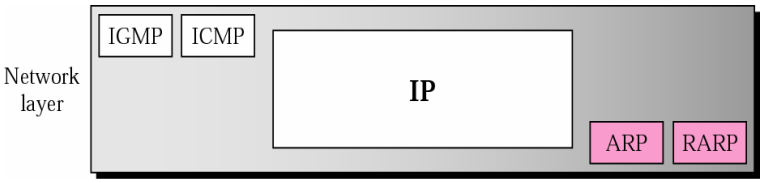


The term address resolution refers to the process of finding an address of a computer in a network. The address is "resolved" using a protocol in which a piece of information is sent by a client process executing on the local computer to a server process executing on a remote computer.

The information received by the server allows the server to uniquely identify the network system for which the address was required and therefore to provide the required address. The address resolution procedure is completed when the client receives a response from the server containing the required address.

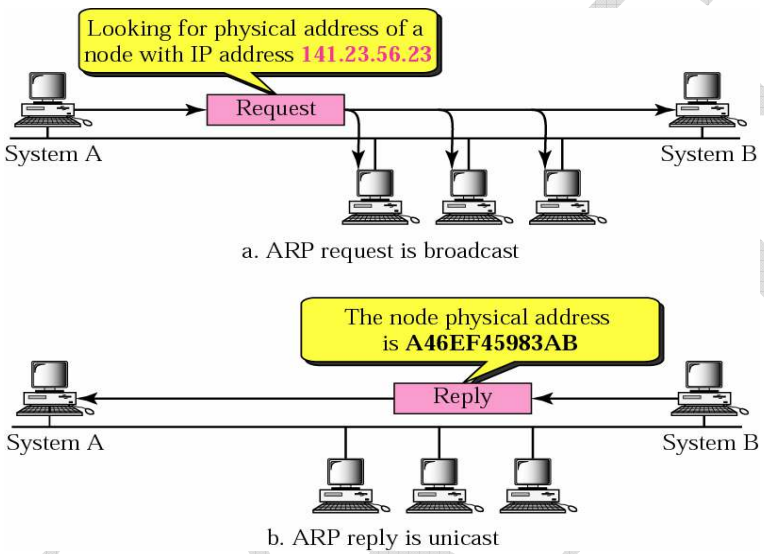
Position of ARP in TCP/IP Protocol Suite

ARP associates an IP address with its physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address that is usually imprinted on the NIC.



Operations

The ARP request packets are broadcast; the RARP reply packets are unicast.



Packet Format

The Packet format for ARP is given below.

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

The fields are as follows

Hardware type (HTYPE): This field specifies the Link Layer protocol type. Example: Ethernet is 1.

Protocol type (PTYPE): This field specifies the upper layer protocol for which the ARP request is intended. For example, Internet Protocol (IPv4) is encoded as 0x0800.

Hardware length (HLEN): Length (in octets) of a hardware address. Ethernet addresses size is 6.

Protocol length (PLEN): Length (in octets) of a logical address of the specified protocol (cf. PTYPE). IPv4 address size is 4.

Operation: Specifies the operation that the sender is performing: 1 for request, 2 for reply.

Sender hardware address (SHA): Hardware (MAC) address of the sender.

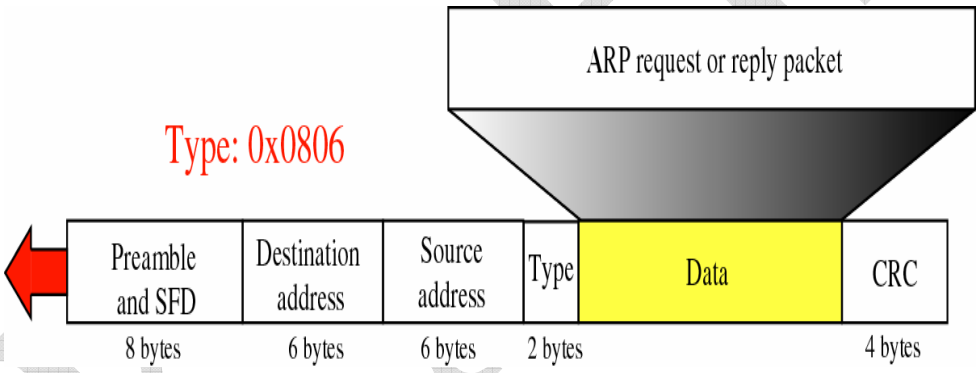
Sender protocols address (SPA): Upper layer protocol address of the sender.

Target hardware address (THA) : Hardware address of the intended receiver. This field is ignored in requests.

Target protocol address (TPA): Upper layer protocol address of the intended receiver.

Encapsulation

ARP packet is encapsulated directly into a data link frame ARP packet encapsulated in an Ethernet frame

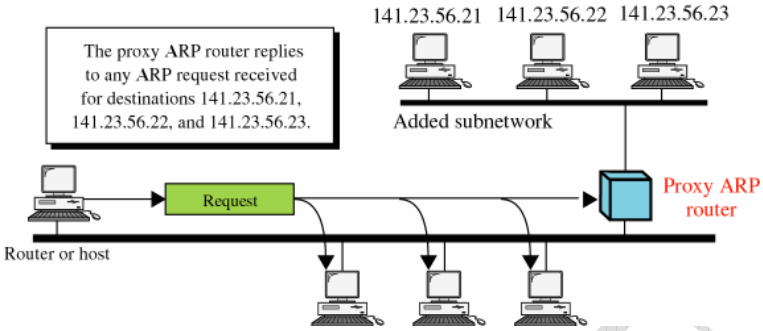


ARP over ATM

ARP is also used when an IP Packet wants to pass over an ATM Network

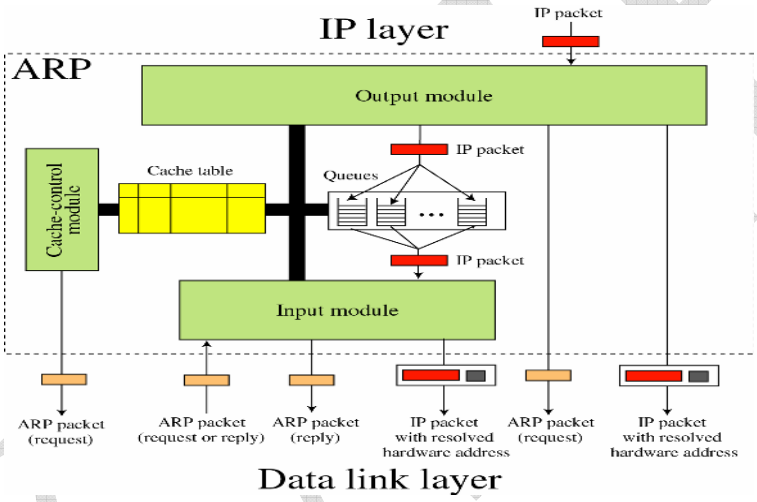
3. PROXY ARP:

ARP that acts on behalf of a set of hosts. Whenever the router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, router sends an ARP reply announcing its own hardware (physical) address later, when the router receives the actual IP packet, it will send the packet to the appropriate host or router



ARP Package

ARP Package includes five components namely Input module, Output module, Cache table, Query and cache control module.



1. Cache table:

When a host or router receives the corresponding physical address for an IP datagram, the address can be saved in the cache table. This address can be used for the datagram's destined for the same receiver within the next few minutes

2. Input Module:

Waiting until an ARP packet (request or reply) arrives. It role is checking the cache table to find an entry corresponding to this ARP packet

Input Module

1. **Sleep until an ARP packet (request or reply) arrives**
2. **Check the cache to find an entry corresponding to the this ARP packet**
3. **If (found)**
    1. **If (the state is PENDING)**

1. **Update the entry**
2. **While the queue is not empty**
    1. **Dequeue one packet**
    2. **Send the packet and the hardware address to date link**
2. **If (the state is RESOLVED)**
    1. **Update the entry**

4. **If (not found)**
    1. **Create an entry**
    2. **Add the entry to the table**
5. **If (the packet is a request)**
    1. **Send an ARP reply**

6. Return

3. **Output Module:**

Waiting for an IP packet from the IP software. Its main role is checking the cache table to find an entry corresponding to the destination IP address of this packet

Output module

1. **Sleep until an IP packet is received from IP software**
2. **Check the cache table to find an entry corresponding to this IP packet**
3. **If (found)**
    1. **If (the state js Resolved)**
        1. **Extract the value of the hardware address from the entry**
        2. **Send the packet and the hardware address to data link layer**
        3. **Return**
    2. **If (the state is PENDING)**
        1. **Enqueue the packet to the corresponding queue**
        2. **Return**
4. **If (not found)**
    1. **Create a queue**
    2. **Enqueue the packet**
    3. **Create a cache entry with state set to PENDING and ATTEMPS set to 1**
    4. **Send an ARP request**

Return

4. Queues:

Holding the IP address while ARP tries to resolve the hardware address

5. Cache-control module

This nodule is responsible for maintaining the cache table. It periodically checking the cache table, entry by entry

Cache-control module

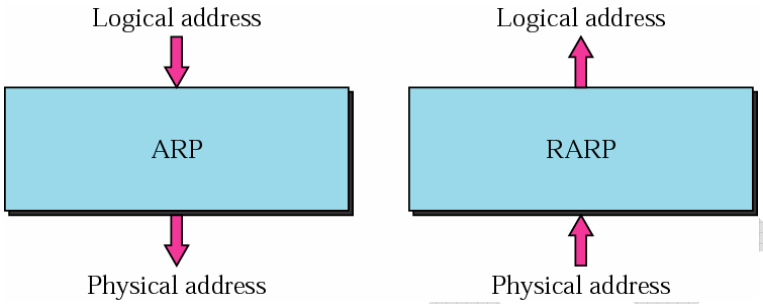
1. **Sleep until the periodic timer matures.**
2. **For every entry in the cache table**
    1. **If(the state is Free)**
        1. **Continue.**
    2. **If(the state is PENDING)**
        1. **Increment the value of attempts by 1.**
        2. **If(attempts greater than maximum)**
            1. **Change the state to FREE**
            2. **Destroy the corresponding queue.**
        3. **3. If(not)**
            1. **Send and ARP request.**
        4. **Continue.**
    3. **If(the state is RESOLVED)**
        1. **Decrement the value of time-out by the value of elapsed time.**
        2. **If(time-out less than or equal to zero)**
            1. **Change the state to FREE.**
            2. **Destroy the corresponding queue**
3. **Return.**

Example:

State	*Queue*	*Attempt*	*Time-out*	*Protocol Addr.*	*Hardware Addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
R	**18**		**900**	**188.11.8.71**	**E34573242ACA**

2. RARP:

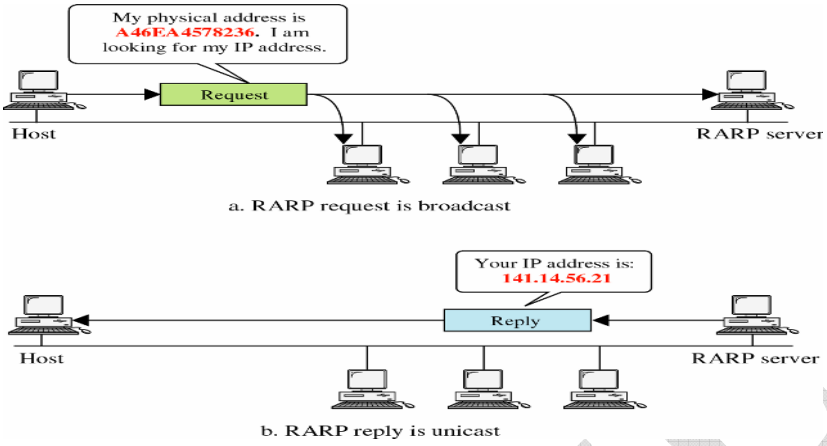
RARP (Reverse Address Resolution Protocol) allows a physical machine in a local area network to request its IP address from a gateway server's Address Resolution Protocol (ARP) table or cache.



To create an IP datagram a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file

Advantages of RARP

1) RARP (Reverse Address Resolution Protocol) allows a physical machine in a local area network to request its IP address from a gateway server's Address Resolution Protocol (ARP) cache or table.
2) A network administrator creates a table in a local area network's gateway router that maps the physical machine (or Media Access Control - MAC address) addresses to corresponding to the Internet Protocol addresses (IP address).
3) When a new machine is set up, its RARP client program requests from RARP server on the router to be sent its IP address.
4) Assuming that an entry has been set up in the router table and the RARP server will return the IP address to the machine which can store it for future use.
5) RARP is available for Fiber Distributed-Data Interface, Ethernet, and Token Ring LANs and ARP (Address Resolution Protocol) performs the opposite function as the RARP: mapping of an IP address to a physical machine address.
6) RARP functionality supports multiple physical network types. All machines receive RARP requests but only those authorized to supply RARP services can respond. Machines supplying RARP services are called RARP servers.. RARP is only used on LANs with a low probability of failure since bootstrapping requires quick responses.



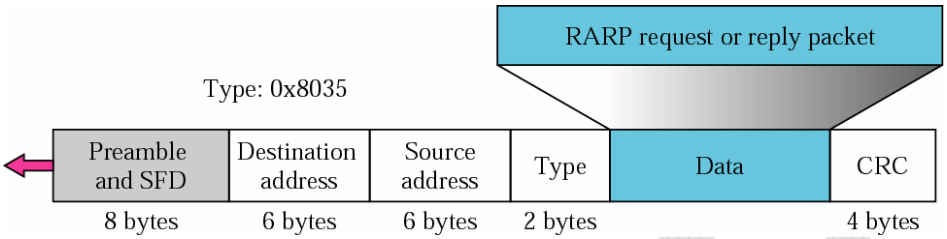
Packet Format

Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

1) **Hardware type -** which specifies hardware interface type for which the sender requires a response.
2) **Protocol type** -which specifies the type of the high-level protocol address the sender has supplied.
3) **Hlen -** Hardware address length.
4) **Plen -** Protocol address length.
5) **Sender hardware address** -HLen bytes in length.
6) **Sender protocol address -** PLen bytes in length.
7) **Target hardware address -** HLen bytes in length.
8) **Target protocol address -** PLen bytes in length.

Encapsulation

A RARP packet is encapsulated directly in to a data link frame. For example, in above fig. shows a RARP packet encapsulated in an Ethernet frame. Note that the type of fields shows that the data carried by a frame is a RARP packet.



RARP Server

All the mappings between the hardware MAC addresses and the IP addresses of the hosts are stored in a configuration file in a host in the network. This host is called the RARP server. This host responds to all the RARP requests. The mapping between MAC addresses and IP addresses is usually stored in a configuration file in the local hard disk in the RARP server. When a RARP server receives a RARP request packet it performs the following steps:

- The MAC address in the request packet is looked up in the configuration file and mapped to the corresponding IP address.
- If the mapping is not found, the packet is discarded.
- If the mapping is found, a RARP reply packet is generated with the MAC and IP address. This packet is sent to the host, which originated the RARP request.

Alternative Solutions to RARP

When a diskless computer is booted, it needs more information in addition to its IP address. It needs to know its subnet mask, the IP address of a router, and the IP address of a name server. RARP cannot provide this extra information. New protocols have been developed to provide this information. The two protocols, BOOTP and DHCP that can be used instead of RARP.

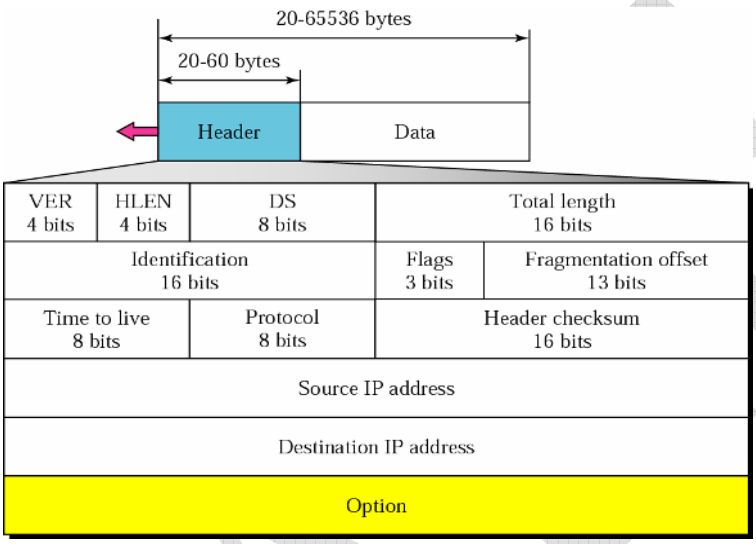
Internet Protocol

The Internet Protocol (IP) is a network-layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is documented in RFC 791 and is the primary network-layer protocol in the Internet protocol suite. Along with the Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols. IP has two primary responsibilities: providing connectionless, best-effort delivery of datagram's through an internetwork; and

providing fragmentation and reassembly of datagrams to support data links with different maximum-transmission unit (MTU) size

4. IP Datagram

A packet in the IP layer is called a datagram, a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery.



The following are the fields in IP Datagram

Version— Indicates the version of IP currently used.

***IP Header Length (*IHL)**—indicates the datagram header length in 32-bit words.

Type-of-Service—Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.

Identification—Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

***Flags*—**Consists of a 3-bit field of which the two low-order (least-significant) bits control

Fragmentation. The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

Fragment Offset—Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

Time-to-Live—Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

***Protocol*—**Indicates which upper-layer protocol receives incoming packets after IP processing is complete.

***Header Checksum*—**helps ensure IP header integrity.

S*ource Address*—Specifies the sending node.

Destination Address—Specifies the receiving node..

***Options*—**Allows IP to support various options, such as security.

***Data*—**Contains upper-layer information.

Routing IP Datagram

If all computers were directly connected on the same physical network, there would be little need for the IP protocol. After all, so far in this description of the protocol, the only job IP has performed has been wrapping the transport layer packet into an IP datagram for transmission by the network level. In reality, an IP datagram sent between two computers on the public network typically passes through many different IP network devices along the way. It is the ability to route IP packets across different physical networks that is the heart of the Internet.

The public network, or Internet, is actually a collection of thousands of individual networks, interconnected together. These interconnections form a mesh network, creating millions of paths between the individual computers on the Internet. Routers are dedicated devices that are the interconnection point for the networks of the world. Routers are responsible for passing IP packets along from the source to the destination, across the various network interconnection points.

Each router that an IP packet passes through is referred to as a hop. In general, as the packet traverses the network, a router is only responsible for getting a packet to the next hop along its path. Routers use the Internet and network layer. Routers need access to the network layer so they can physically receive packets. The network layer then passes the IP datagram up to the router IP layer. The router processes the destination address contained in the IP header and determines which device the send the IP packet on to, typically another router. The transport and user level data is not needed and is not unpacked from the IP datagram. This allows routers to function very quickly, as they are able to unpack the necessary information from the IP packet using specially designed hardware.

Routing Protocols

Routers are responsible for routing IP packets between a source and destination address. Typically, each router is responsible for only getting a packet to the next router along the path. As such, a router only needs to know the addresses of the routers to which it is directly connected. It also needs to know which connected router should be used for forwarding a packet. When the router examines the IP

address of an incoming datagram, it accesses a database or table to determine which router should form the next hop in the path. Routers use various protocols to communicate with each other in order to set up the tables used to route packets.

Some common routing protocols include:

- Router Information Protocol (RIP)
- Open Shortest Path First (OSPF)
- Interior Gateway Routing Protocol (IGRP)
- Enhanced IGRP (EIGRP)
- Border Gateway Protocol (BGP)
- Intermediate System to Intermediate System (IS-IS)
- Constrained Shortest Path First (CSPF)

While routers operate on packets at the Internet layer, they also use transport layer services such as UDP and TCP to communicate with each other to build routing tables.

5. Fragmentation

If a router receives an IP packet that is too large for the network onto which the packet is being forwarded IP will fragment the original packet into smaller packets that will fit on the downstream network. When the packets arrive at their final destination, IP at the destination host reassembles the fragments into the original payload.

This process is referred to as fragmentation and reassembly. Fragmentation can occur in environments that have a mix of networking technologies, such as Ethernet and Token Ring.

The fragmentation and reassembly works as follows:

1) When an IP packet is sent by the source, it places a unique value in the Identification field.
2) The IP packet is received at the router. The IP router notes that the maximum transmission unit (MTU) of the network onto which the packet is to be forwarded is smaller than the size of the IP packet.
3) IP fragments the original IP payload into fragments that will fit on the next network. Each fragment is Sent with its own IP header which contains

IP Header Field	Function
Source IP Address	The IP address of the original source of the IP datagram.
Destination IP Address	The IP address of the final destination of the IP datagram.
Identification	Used to identify a specific IP datagram and to identify all fragments of a specific IP datagram if fragmentation occurs.
Protocol	Informs IP at the destination host whether to pass the packet up to TCP, UDP, ICMP, or other protocols.
Checksum	A simple mathematical computation used to verify the integrity of the IP header.
Time to Live (TTL)	Designates the number of networks on which the datagram is allowed to travel before being discarded by a router. The TTL is set by the sending host and is used to prevent packets from endlessly circulating on an IP internetwork. When forwarding an IP packet, routers are required to decrease the TTL by at least one.

The original Identification field identifies all fragments that belong together. The More Fragments Flag indicates that other fragments follow. The More Fragments Flag is not set on the last fragment, because no other fragments follow it.

The Fragment Offset field indicates the position of the fragment relative to the original IP payload. When the fragments are received by IP at the remote host, they are identified by the Identification field as belonging together. The Fragment Offset is then used to reassemble the fragments into the original IP payload.

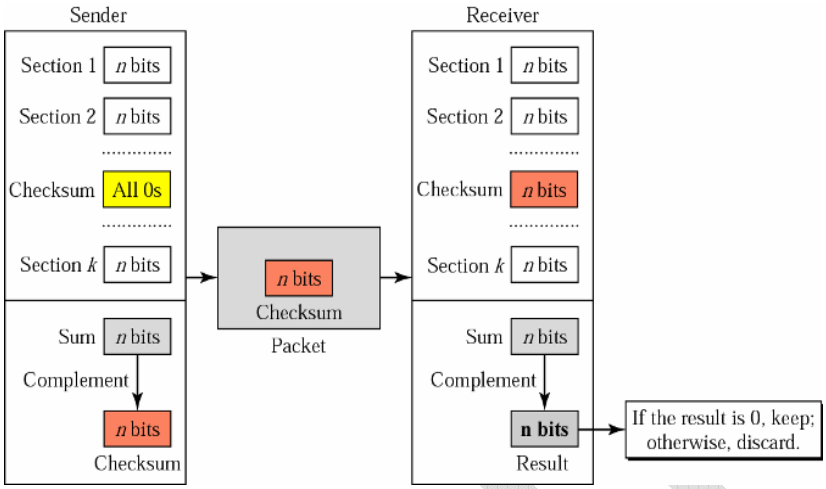
Check-Sum

The error detection method used by most TCP/IP protocols is called the checksum. The checksum protects against the corruption that may occur during the transmission of a packet. It is redundant information added to the packet.

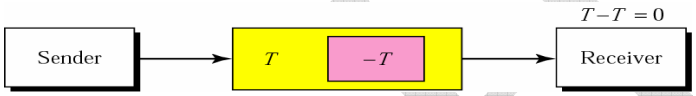
To create the checksum the sender does the following:

1) The packet is divided into k sections, each of n bits.
2) All sections are added together using 1's complement arithmetic.
3) The final result is complemented to make the checksum.

Checksum Concept:

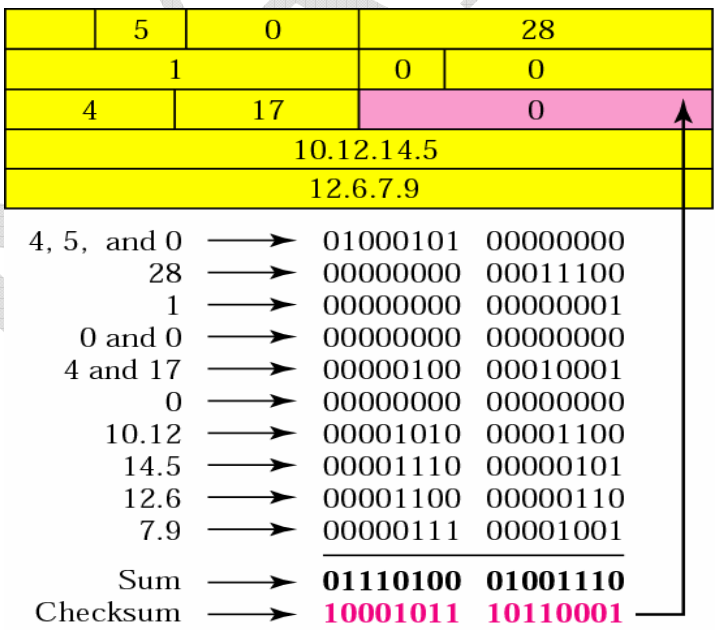


Checksum in one's complement arithmetic



Example:

The following shows an example of a checksum calculation for an IP header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.

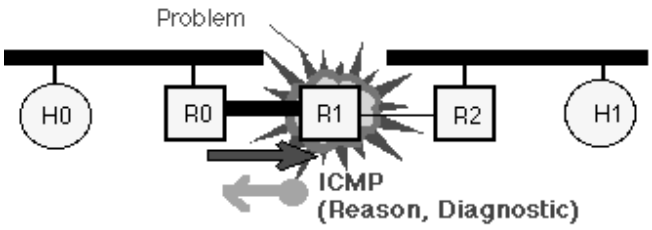


.

5. Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) protocol is classic example of a client server application. The ICMP server executes on all IP end system computers and all IP intermediate systems (i.e routers). The protocol is used to report problems with delivery of IP datagrams within an IP network. It can be sued to show when a particular End System (ES) is not responding, when an IP network is not reachable, when a node is overloaded, when an error occurs in the IP header information, etc. The protocol is also frequently used by Internet managers to verify correct operations of End Systems (ES) and to check that routers are correctly routing packets to the specified destination address.

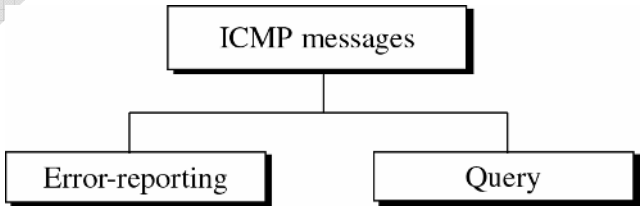
ICMP messages generated by router R1, in response to message sent by H0 to H1 and forwarded by R0. This message could, for instance be generated if the MTU of the link between R0 and R1 was smaller than size of the IP packet, and the packet had the Don't Fragment (DF) bit set in the IP packet header.The ICMP message is returned to H0, since this is the source address specified in the IP packet that suffered the problem.



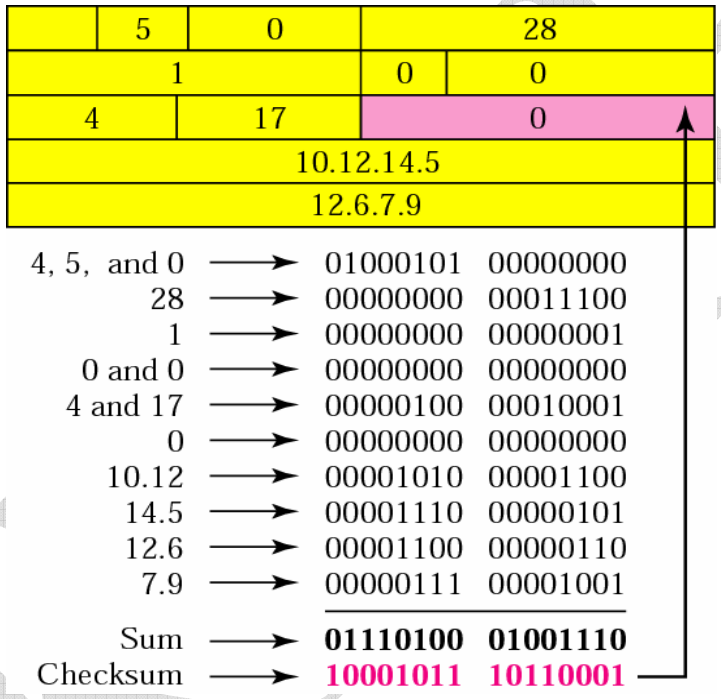
ICMP Messages

ICMP Messages are used by IP to send error and control messages ICMP uses IP to send messages. It does not report errors on ICMP messages. ICMP message are not required on datagram checksum errors. There are two types of messages namely

1) Error reporting
2) Query messages.

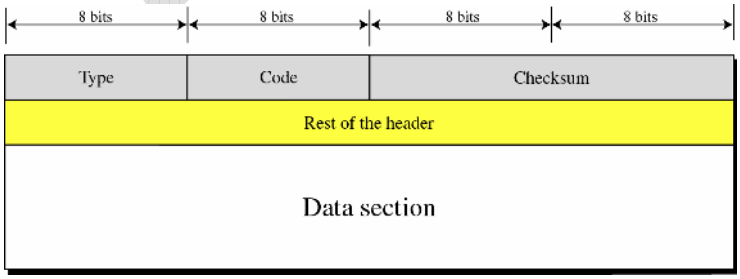


Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router solicitation or advertisement



General format of ICMP messages

An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all.



Type (8): specifies the type of ICMP message

Code (8): used to specify parameters of the message that can be encoded in a few bits

Checksum (16): hecksum of the entire ICMP message

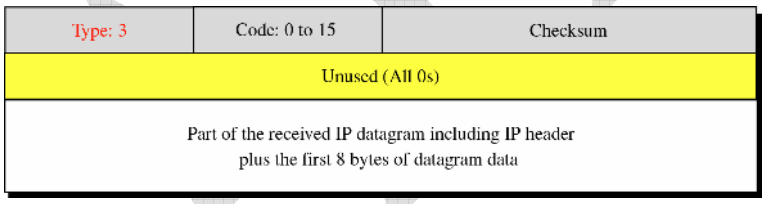
ERROR REPORTING

ICMP always reports error messages to the original source ICMP error messages report error conditions Typically sent when a datagram is discarded Error message is often passed from ICMP to the application program ICMP error essages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)

.

a) Destination-unreachable

Destination-unreachable messages with codes 2 or 3 can be created only by the destination host. Other destination-unreachable messages can be created only by routers. A router cannot detect all problems that prevent the delivery of a packet



Code 0 Net Unreachable

Code 1 Host Unreachable

Code 2 Protocol Unreachable

Code 3 Port Unreachable

Code 4 Fragmentation needed & Don't Fragment was set

Code 5 Source Route failed

Code 6 Destination Network Unknown

Code 7 Destination Host Unknown

Code 8 Source Host Isolated

Code 9 Communication Destination Network is Administratively Prohibited

Code 10 Communication Destination Host is

Administratively Prohibited

Code 11 Destination Network Unreachable for Type of Service

Code 12 Destination Host Unreachable for Type of Service

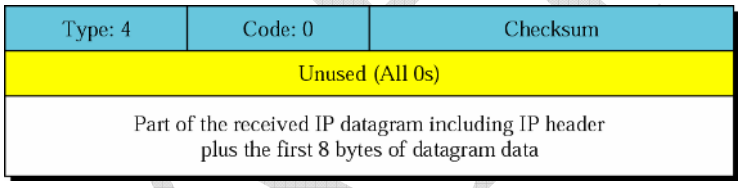
Code 13 Communication Administratively Prohibited

Code 14 Host Precedence Violation

Code 15 Precedence Cutoff Violation

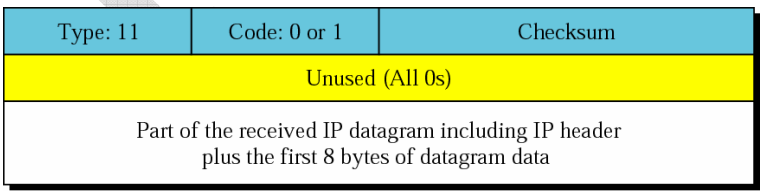
b) Source-quench

A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host. The source must slow down the sending of data grams until the congestion is relieved. One source-quench message should be sent for each datagram that is discarded due to congestion.



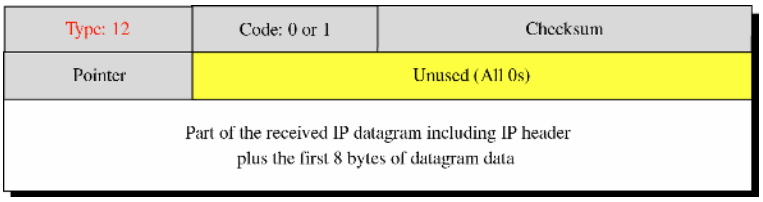
c) Time Exceed

Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source. When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source. In a time-exceeded message, code 0 is used only by routers to show that the value of the time-to-live field is zero. Code 1 is used only by the destination host to show that not all of the fragments have arrived within a set time.



d) Parameter Problem

A parameter-problem message can be created by a router or the destination host.

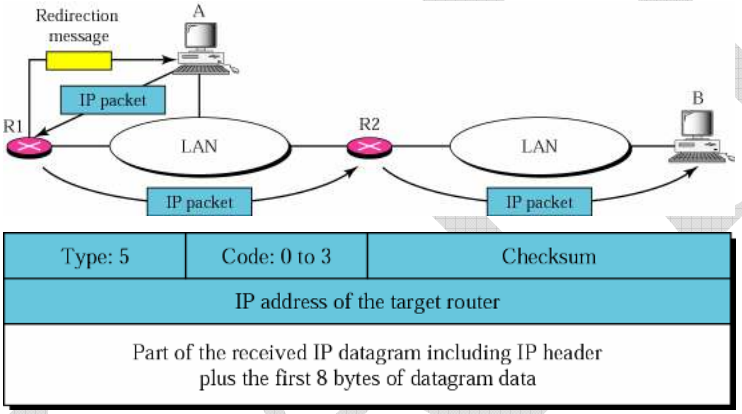


Code 0: Main header problem

Code 1: Problem in the option field

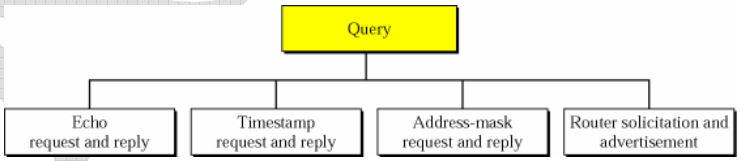
e) Redirection

A host usually starts with a small routing table that is gradually augmented and updated. One of the tools to accomplish this is the redirection message. A redirection message is sent from a router to a host on the same local network.



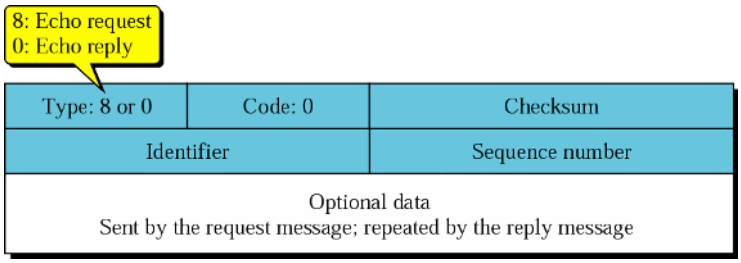
QUERY MESSAGES

ICMP can also diagnose some network problems through the query messages, a group of four different pairs of messages. In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.



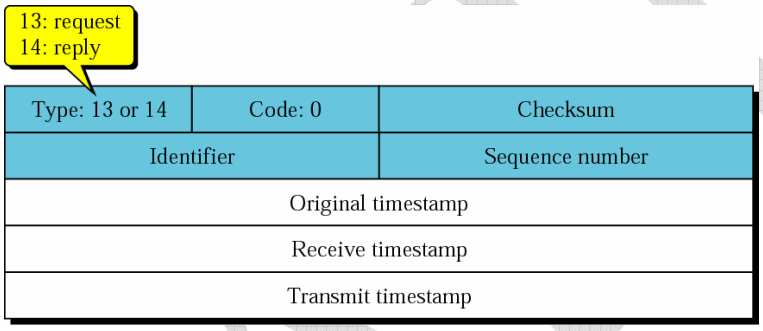
a) Echo Request and Reply:

An echo-request message can be sent by a host or router. An echo-reply message is sent by the host or router which receives an echo-request message. Echo-request and echo-reply messages can be used by network managers to check the operation of the IP protocol. Echo-request and echo-reply messages can test the reach ability of a host. This is usually done by invoking the ping command.



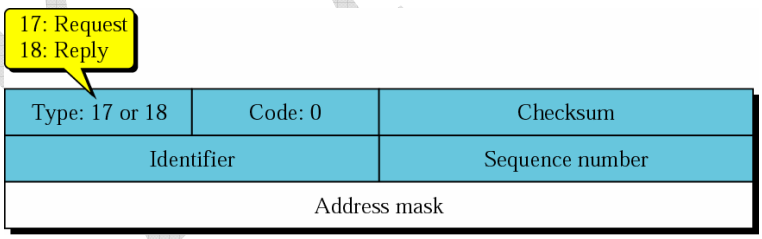
b) Timestamp Request and Reply

Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine even if their clocks are not synchronized. The timestamp-request and timestamp-reply messages can be used to synchronize two clocks in two machines if the exact one-way time duration is known.



c) Address Mask Request & Address Mask Reply

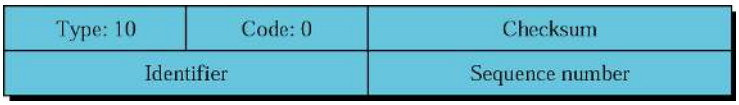
A booting computer to determine the subnet mask in use on the local network uses the Address Mask Request ICMP Type 17. An intermediary device or computer acting as an intermediary device will reply with a Type 18 ICMP Address Mask Reply ICMP.



d) **Router-solicitation message**

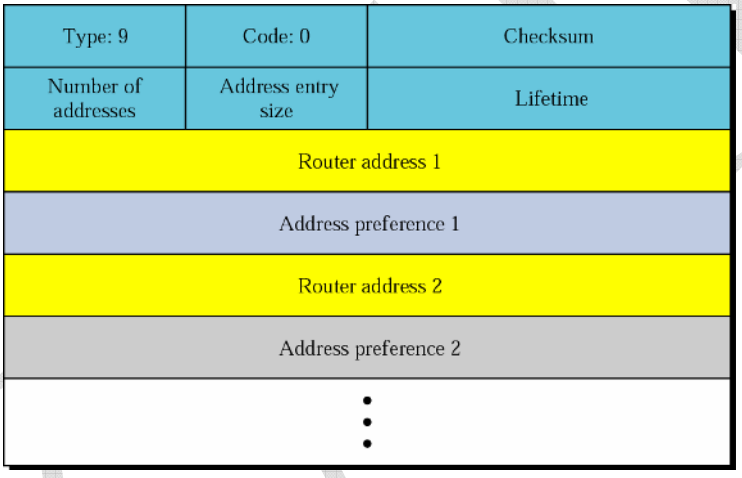
Router discovery uses Internet Control Message Protocol (ICMP) router advertisements and router solicitation messages to allow a host to discover the addresses of operational routers on the subnet. Hosts must discover routers before they can send IP datagrams outside their subnet. Router discovery allows a host to discover the addresses of operational routers on the subnet. Each router periodically multicasts a router advertisement from each of its multicast interfaces, announcing the IP address of that

interface. Hosts listen for advertisements to discover the addresses of their neighboring routers. When a host starts, it can send a multicast router solicitation to ask for immediate advertisements.



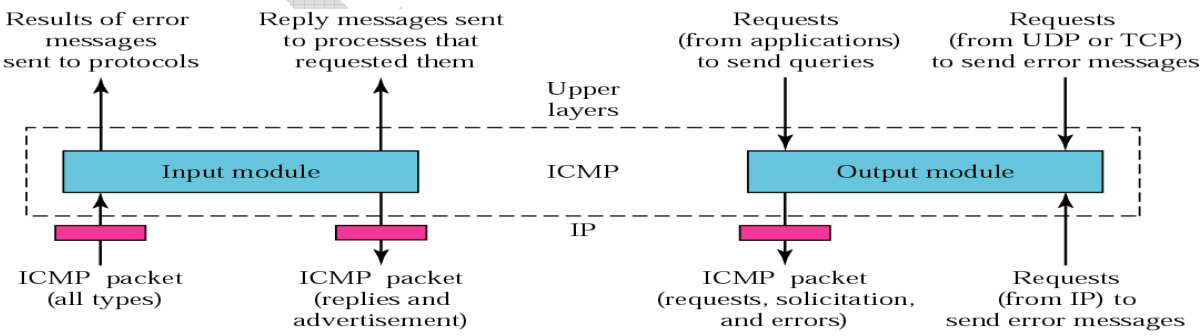
e) Router advertisement message

Router advertisement messages include a preference level and a lifetime field for each advertised router address. The preference level specifies the router's preference to become the default router. When a host chooses a default router address, it chooses the address with the highest preference. You can configure the preference level with the priority statement. The lifetime field indicates the maximum length of time that the advertised addresses are to be considered valid by hosts in the absence of further advertisements..



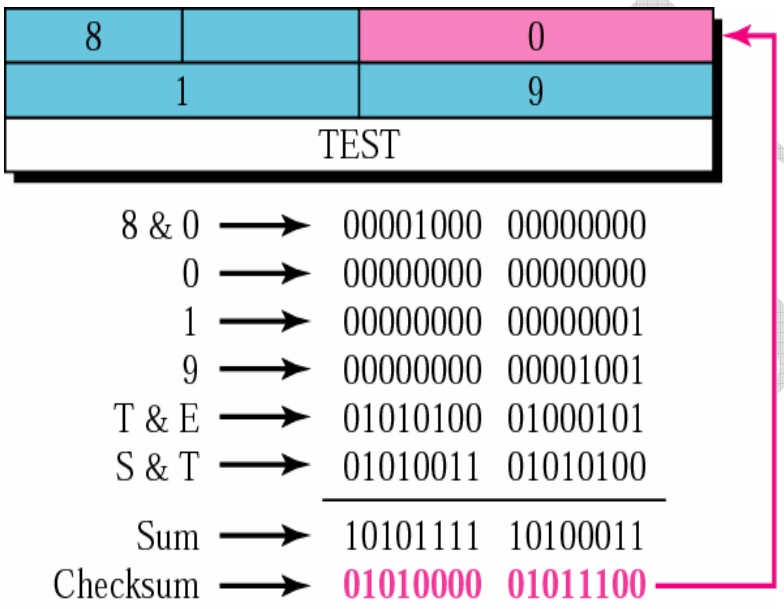
ICMP Package

To give an idea of how ICMP can handle the sending and receiving of ICMP messages, we present our version of an ICMP package made of two modules: an input module and an output module



Checksum

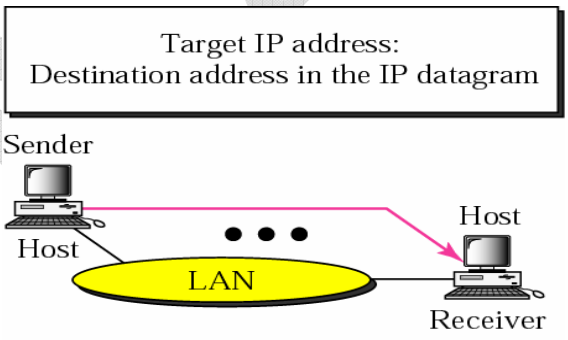
In ICMP the checksum is calculated over the entire message (header and data). The below figure shows an example of checksum calculation for a simple echo-request message. We randomly chose the identifier to be 1 and the sequence number to be 9. The message is divided into 16-bit (2-byte) words. The words are added together and the sum is complemented. Now the sender can put this value in the checksum field.



Example 1:

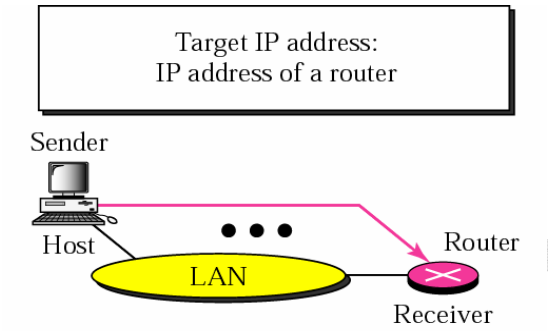
Four cases of ARP:

Case 1:



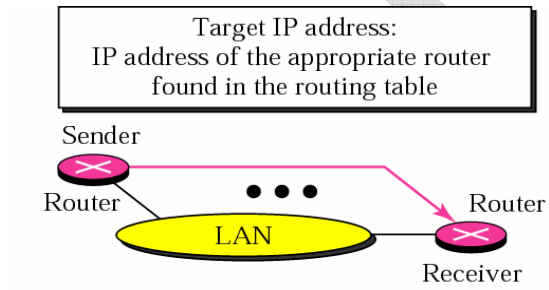
Case 1. A host has a packet to send to another host on the same network.

Case 2:



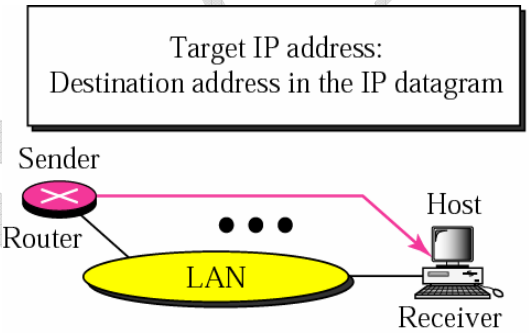
Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to a router.

Case 3 :



Case 3. A router receives a packet to be sent to a host on another network.
It must first be delivered to the appropriate router.

Case 4:



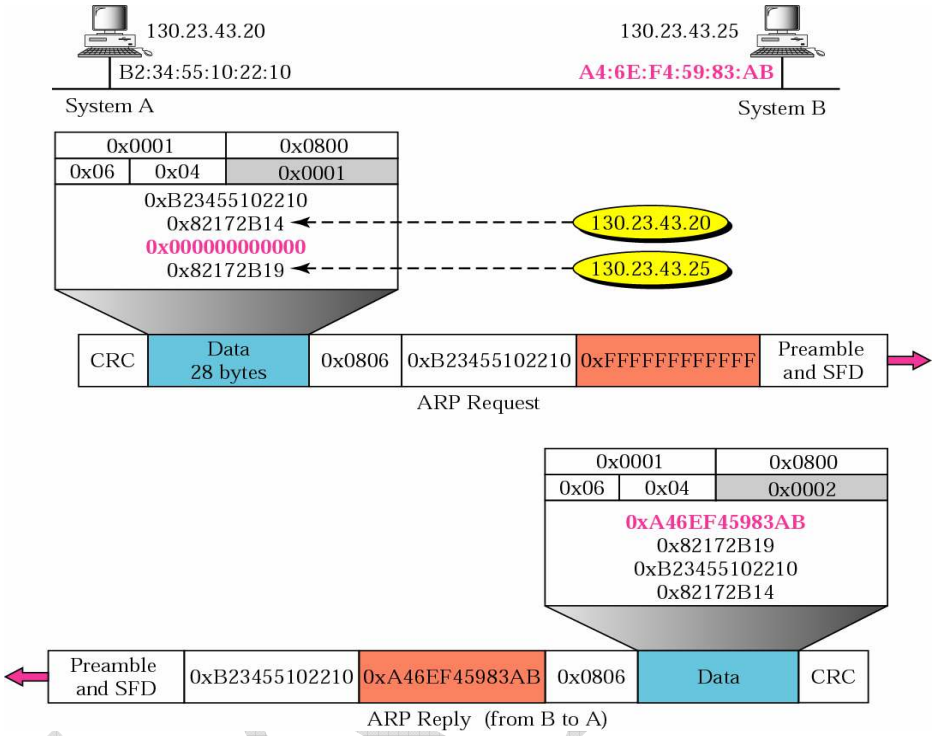
Case 4. A router receives a packet to be sent to a host on the same network.

Example 2:

Problem: A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB (which is

unknown to the first host). The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution :



The above figure shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses. Also note that the IP addresses are shown in hexadecimal.

Example 3:

Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine even if their clocks are not synchronized

Sending time = value of receive timestamp - value of original Timestamp

Receiving time = time the packet returned - value of transmit Timestamp

Round-trip time = sending time + receiving time

Given the following information:

Value of original timestamp: 46

Value of receive timestamp: 59

Value of transmit timestamp: 60

Time the packet arrived: 67

Sending time = 59 - 46 = 13 milliseconds

Receiving time = 67 - 60 = 7 milliseconds

Round-trip time = 13 + 7 = 20 milliseconds

Time diff= receive timestamp -(original timestamp Field + one-way time duration)

Possible Questions

Two Mark Questions

1. What is a physical Address?
2. What is the use of Logical Address?
3. List the areas where ARP is used.
4. What is an IP datagram?
5. Define a checksum.

Six Mark Questions

1. What is address mapping? Discuss about the static and dynamic binding of mapping.
2. What is ARP? Discuss about its packet format and its operation in detail
3. Write a detail note on ATMARP and its operations.
4. Discuss about IP Datagram in detail
5. Explain about IP datagram Options in details
6. Discuss about IP Checksum process in detail
7. Discuss IP data fragmentation in detail
8. Compare TCP/IP model with OSI Reference Model
9. Discuss about ICMP Query Messages.
10. Explain about ICMP Message Reporting Messages

Unit - II

S.No	Question	Choice1	Choice2	Choice3	Choice4	Answer
1	A loose source route option is similar to _____ option.	Time Stamp option	Strict Source route option	End of option	Both a and b	Strict Source route option
2	The _____ contains the data that specific option requires.	Length field	data field	Width field	No field	data field
3	HLEN stands for _____	Header Length	Heading Length	Highlight Length Enable Network	None of the above	Header Length
4	_____ is a 3 bit sub field ranging from 0 to 7	Precedence	TOS Bits	Code point	protocol	Precedence
5	_____ is a local address.	physical	logical	network	IP	physical
6	_____ attaches to two or more physical networks and forwards IP datagrams	Router	Hub	Bridge	None	Router
7	Internet routers could be portioned into _____ and _____	MILNET and ARPANET	Core and Non core	Individual and Non individual	None	Core and Non core
9	_____ travels from routers to routers	Text	Numbers	Datagrams	Characters.	Datagrams
10	_____ is used to measure distance	Core	Vector	Dynamic	Hops	Hops
11	___is a group of network controlled by single administrative autthority	Non autonomous system	Autonomous system	Single system	None	Autonomous system
12	_____ forms the central interconnect for an internet	Base header	Application Gateway	Backbone network	Local Area Network	Backbone network
13	_____ is used to pass information between two autonomous system	EGP	IGP	DGP	CGP	EGP
14	BGP evolves _____ versions	3	2	4	1	4
15	__ protocol is used to propagate routing information inside a autonomous system	IGP	RIP	RFC	None	RIP
16	RIP messages can be broadly classified into _____ types	3	2	4	5	2
17	_____ reference to processor choosing the path to send packets	Router	Hub	Bridge	None	Router
18	Routers exchange _____ to initialize their network	OSFP	OSPF	OPSF	OFSP	OSPF
19	OSPF sends _____ message to test neighbour reach ability	Hello	Check	Transmit	None	Hello
20	Command1 terns on the _____ mode	update	Request	Trace	None	Trace
21	Final technique for solving the slow convengaice problem is called _____	Triggered updates	Poison reverce	Both a and b	Null	Poison reverce
22	Path information in BGP allows the receiver to check for _____	Routing loops	Routing algorithms	Both A and B	None	Routing loops
23	BGP periodically exchange ____message to text network connectivity	OPEN	EXCHANGE	CHECK	KEEPALIVE	KEEPALIVE
24	Command 10 in RIP is _____	Update Request	Update Response	Update acknowledgement	None	Update Response
25	_____ used by routing protocols to combine multiple destination same hope	Route aggregation	Route alert	Route integration	None	Route aggregation
26	The _____is one of the necessary protocols that is involved In multicasting	ICMP	IGMP	TCP	IP	IGMP
27	_____ defines the amount of time in which a query must be answered	Maximum response time	Minimum response time	Response time	Check sum	Maximum response time
28	To prevent unnecessary traffic , IGMP uses a _____ strategy	Maximum response time	Minimum response time	Delayed response	response	Delayed response
29	Well known port numbers are less than_____	255	1024	125	206	1024
30	_____are the ports ranging from 0 to 1023, are assigned and controlled by LCANN	Registered ports	Dynamic ports	Static ports	Well known ports	Well known ports
31	The combination of _____ and _____, called socket address	IP Address, Port Number	IP Header, Port Number	Socket Number, Port Number	IP Header, Socket Number	IP Address, Port Number
32	TCP offers _____ services	Full duplex	Half Duplex	Both a and b	Multi duplex	Full duplex
33	A packet in a TCP is called _____	Frame	Segment	Data	Bits	Segment
34	_____ is a machine that goes through a limited number of states	State machine	Finite State machine	Limited state machine	None of the above	Finite State machine
35	To accomplish flow control, TCP uses a _____ protocol	ICMP	IGMP	Sliding window	SNMP	Sliding window
36	A _____ is packet sent by a router to the source to inform it of congestion	Choke point	Back pressure	Implicit Signaling	Explicit signaling	Choke point
37	In _____ routing router needs to construct a shortest path tree for each group	Unicast routing	Multicast routing	Multicast link state routing	Multicast distance vector routing	Multicast routing
38	_____ protocol is a group shared protocol that uses a core as the root of the tree	ICMP	DVMRP	CBT protocol	IGMP	CBT protocol
39	PIM-SM is used in a sparse multicast environment such as _____	WAN	LAN	MAN	Both a and b	WAN
40	The value of group address is _____ for a general query message	2	3	0	4	0
41	The port ranging from 1024 to 49151, assigned or controlled by ICANN, known as	Dynamic ports	Static ports	Well known ports	Registered ports	Registered ports
42	The UDP packets, called used data grams have a fixed size header of _____ bytes	6	8	32	64	8
43	The connection establishing in TCP is called _____	3-way hand shaking	4-way hand shaking	2-way hand shaking	1-way hand shaking	3-way hand shaking
44	SYN flooding attack belongs to a group of security attacks known as _____ attack	Denial of service	Mosquerade	Replay	Null	Denial of service
45	One of the algorithm used in TCP congestion control is _____	Fast start	Slow start	RFC algorithm	None	Slow start
46	_____is a client/server protocol designed to provide information for a disk less computer	BOOTP	RARP	ICMP	ARP	BOOTP
47	Bootstrap protocol is an _____ layer program.	network	transport	application	physical	application
48	The host that can be used as a relay to operate at the application layer is _____	BOOTP server	BOOT client	relay agent	none	relay agent
49	The relay agent knows the _____address of a BOOT server	multicast	unicast	broadcast	network address	unicast
50	The ID which is randomly choosen for each connection involving BOOTP is_____	transaction ID	net ID	host id	BOOTP id	transaction ID
51	BOOTP uses _____which does not provide error control	TCP	UDP	TFTP	ARP	UDP
52	_____ is a 8 bit field defining the maximum number of hops the packet can travel	operation code	hardware type	hardware length	hop count	hop count
53	_____ is a 4 bit field containing the IP address of a router	gate way IP address	client IP address	server IP address	none	gate way IP address

54	_____ provides static and dynamic address allocation that can be manual or automatic.	BOOTP	DHCP	TFTP	RARP	DHCP
55	_____ is backward compatible with BOOTP	DHCP	BOOTP	UDP	ARP	DHCP
56	DHCP server issues a_____for a specific period of time	time stamp	time slice	lease	none	lease
57	A server reply can be _____	broadcast	unicast	multicast	broadcast , unicast	broadcast , unicast
58	_____is a static configuration protocol	BOOTP	TFPT	UDP	RARP	BOOTP
59	_____ is a client server application on the internet with the unique user friendly name	DNS	DDNS	FQDN	PQDN	DNS
60	A _____ server gets its information from a primary server	root server	DNS server	BOOTP server	secondary server	secondary server
61	_____ server creates maintains and updates information about its zone	secondary	primary	root	BOOTP	primary

SYLLABUS

Unit III: Group management – IGMP message – IGMP operation – process to process communication – UDP operation – TCP services – Flow control.

1. Group Management

Internet Group management protocol (IGMP), a multicasting protocol in the internet protocols family, is used by IP hosts to report their host group memberships to any immediately neighboring multicast routers. IGMP messages are encapsulated in IP datagrams, with an IP protocol number of 2. IGMP has versions IGMP v1, v2 and v3

IGMPv1: Hosts can join multicast groups. There were no leave messages. Routers were using a time-out based mechanism to discover the groups that are of no interest to the members.

IGMPv2: Leave messages were added to the protocol. Allow group membership termination to be quickly reported to the routing protocol, which is important for high-bandwidth multicast groups and/or subnets with highly volatile group membership.

IGMPv3: Major revision of the protocol. It allows hosts to specify the list of hosts from which they want to receive traffic from. Traffic from other hosts is blocked inside the network. It also allows hosts to block inside the network packets that come from sources that sent unwanted traffic.

The variant protocols of IGMP are:

DVMRP: Distance Vector Multicast Routing Protocol.

IGAP: IGMP for user Authentication Protocol.

RGMP: Router-port Group Management Protocol.

Protocol Structure -IGMP (Internet Group Management Protocol)

There are basically 5 types of messages in the IGMP that must be implemented in IGMP for the IGMP v3 functional properly and be compatible with previous versions:

0x11: membership query

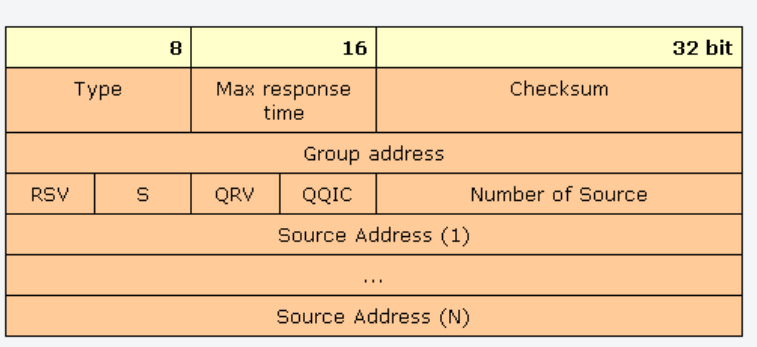
0x22: version 3 membership report

0x12: version 1 membership report

0x16: version 2 membership report

0x17 version 2 leave group

As an example, the message format for 0x11 (membership query) is displayed as follows



8		16		32 bit
Type		Max response time		Checksum
Group address				
RSV	S	QRV	QQIC	Number of Source
Source Address (1)				
...				
Source Address (N)				

Type -- The message type: 0x11 (Membership query).

Max Response Time -- Used only in Membership query messages. Specifies the maximum time allowed before sending a responding report in units of 1/10 second. In all other messages, it is set to 0 by the sender and ignored by the receiver.

Checksum -- The checksum for messages errors

Group Address -- The Group address is set to 0 when sending a general query. It is set to the group address being queried, when sending a group specific query or group-and-source-specific query. In a membership report of a leave group message, it holds the IP multicast group address of the group being reported or left.

Group Record -- A block of fields containing information about the sender's membership in a single multicast group, on the interface from which the report is sent.

IGMP Messages

There are three message types used in IGMP. The IGMP 'type' field is set to the following values for each message type [in hex]:

MEMBERSHIP QUERY [0x11]

Membership Query messages are used by multicast enabled routers running IGMP to discover which hosts on attached networks are members of which multicast groups. Membership Query messages are sent to the 'all-systems' multicast group address of 224.0.0.1.

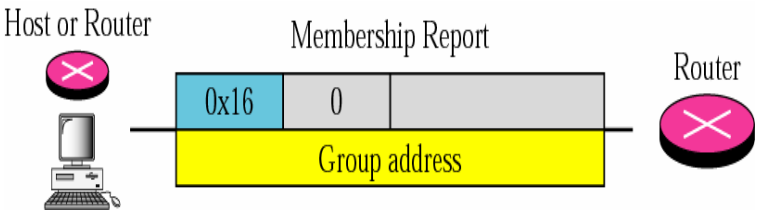
There are two types of Membership Queries:

General Query - used to learn which groups have members on an attached network.

Group-Specific Query - used to learn if a specific group has any members on an attached network.

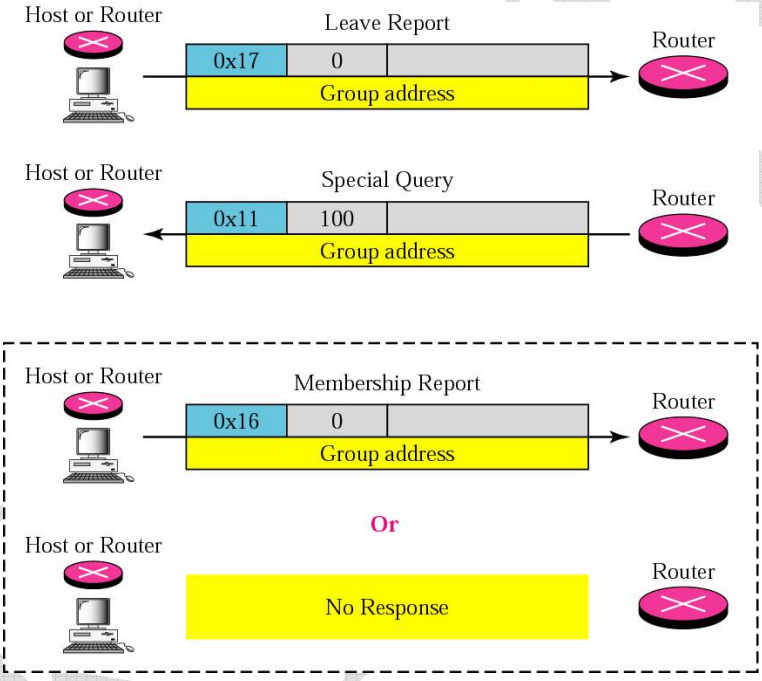
MEMBERSHIP REPORT (v1/v2) [0x12 / 0x16]

A membership report message is sent by a host whenever it joins a multicast group, and when responding to Membership Queries sent by an IGMP router that is functioning as a Querier.



LEAVE GROUP [0x17]

This message is sent when a host leaves a multicast group. This message is sent to the 'all-routers' multicast address of 224.0.0.2. The router then sends out a group-specific membership query to the network to verify if the last member of a group has left.



2. Process to process Communication

Traditionally, the TCP/IP protocol suite has specified two protocols for the transport layer: UDP and TCP. We studied UDP in Chapter11; we will study TCP in this chapter. A new transport-layer protocol called SCTP is now in use by some implementations and will be discussed in Chapter 13.

Figure 12.1 shows the relationship of TCP to the other protocols in the TCP/IP protocol suite. TCP lies between the application layer and the network layer and serves as the intermediary between the application programs and the network operations.

TCP, like UDP, is a process-to-process (program-to-program) protocol. TCP, therefore, like UDP, uses port numbers. Unlike UDP, TCP is a connection-oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow- and error-control mechanism at the transport level.

In brief, TCP is called a connection-oriented reliable transport protocol. It adds connection-oriented and reliability features to the service of IP.

TCP SERVICES

Before discussing TCP in detail, let us explain the services offered by TCP to the processes at the application layer.

Process-to-Process Communication

Like UDP, TCP provides process-to-process communication using port numbers Table lists some well-known port numbers used by TCP.

Table 12.1 Well-known ports used by TCP

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

Example 1

As we said in chapter 11, in UNIX, the well-known ports are stored in a file called /etc/services. Each line in this file gives the name of the server and the well-known port number. We can use the grep utility to extract the line corresponding to the desired application. The following shows the ports for FTP.

Stream Delivery Service

TCP, unlike UDP, is a stream-oriented protocol. IN UDP, a process (an application program) sends message, with predefined boundaries, to UDP for delivery. UDP adds its own header to each of this message and delivers it to IP for transmission. Each message from the process is called a user datagram, and becomes, eventually, one IP datagram. Neither IP nor UDP recognizes any relationship between the datagrams.

TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes. TCP creates an environment in which the two process seem to be connected by an imaginary "tube" that carries their data across the Internet. This imaginary environment is depicted in Figure 12.2. The sending process produces (writes to) the stream of bytes and the receiving process consumes (reads from) them.

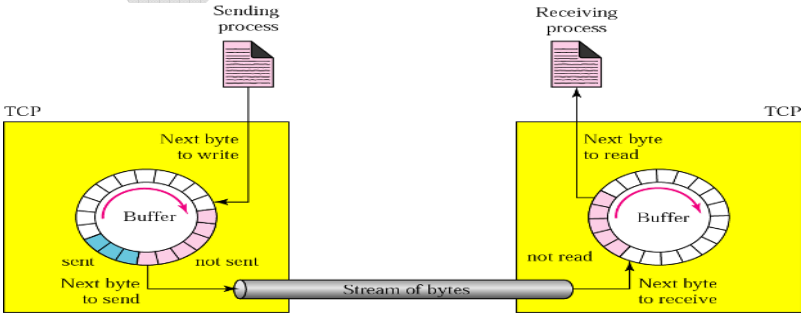
Figure 12.2 Stream delivery

Sending and Receiving Buffers

Because the sending and the receiving process may not write or read at the same speed, TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction. (We will see later that these buffers are also necessary for flow- and error-control mechanism used by TCP). One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure 12.3. For simplicity, we have shown two buffers of 20 bytes each; normally the buffer are hundreds or thousands of bytes, depending on the implementation. We also show the buffers as the same size, which is not always the case.

The figure shows the movement of the data in one direction. At the sending site, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer). The gray area holds bytes that have been sent but not yet acknowledged. TCP keeps these bytes in the buffer until it receives an acknowledgement. The colored area contains bytes to be sent by the sending TCP. However, as we will see later in this chapter, TCP may be able to send only part of this colored section.

Figure 12.3 Sending and Receiving buffers



The operation of the buffer at the receiver site is simpler. The circular buffer is divided into two areas (shown as white and colored). The white contain empty chambers to be filled by the receiving process. When a byte is read by the receiving process, the chambers is recycled and added to the pool of empty chambers.

Segments

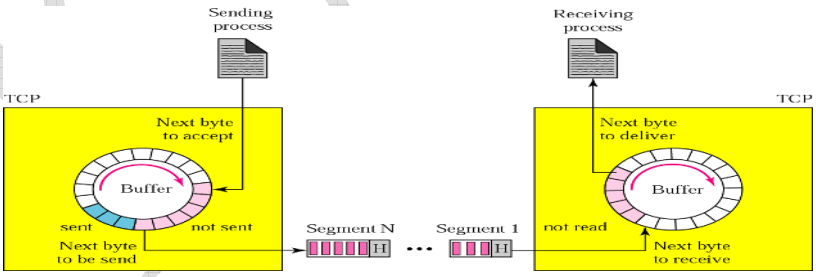
Although buffering handles the disparity between the speed of the producing and consuming process, we need one more step before we can send data. The IP layer, as a service provider for TCP, needs to send data in packets, not as a stream of bytes. At the transport layer, TCP groups a number of bytes together into a packet called segment. TCP adds a header to each segment (for control purpose) and delivers the segment to the IP transmission. The segments are encapsulated in an IP datagram and transmitted. This entire operation is transparent to the receiving process. Later we will see that segments may be received out of order, lost, or corrupted and resent. All of these are handled by TCP with the receiving process unaware of any activities. Figure 12.4 shows how segments are created from the bytes in the buffers.

Note that the segments are not necessary the same size. In the figure, for simplicity, we show one segment carrying 3 bytes and the other carrying 5 bytes. In reality segments carry hundreds if not thousands of bytes.

Full-Duplex Communication

TCP offers **full-duplex service,** where date can flow both directions at the same time. Each TCP then has a sending and the receiving buffer and segments move in both directions.

Figure 12.4 TCP Segments



Connection-Oriented Service

TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at the site B, the following occurs.

1. The two TCPs establish a connection between them.
2. Data are exchanged in both directions.

3. The connection is terminated.

Note that this is a virtual connection, not a physical connection. The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost, or corrupted, and then resent. Each may use a different path to reach the destination. There is no physical connection. TCP creates a stream-oriented environment in which it accepts the responsibility of delivering the bytes in order to the other site. The situation is similar to creating a bridge that spans multiple islands ans passing all of the bytes from one island to another in one single connection. We will discuss this feature later in the chapter.

Reliable Service.

TCP is a reliable transport protocol. It uses an acknowledgement mechanism to check the safe and sound arrival of data. We will discuss this feature further in the section on error control.

TCP FEATURES

To provide the services mentioned in the previous section, TCP has several features that are briefly summarized in this section and discussed later in detail.

Numbering System

Although the TCP software keeps track of the segments being transmitted or received, there is no field for a segment number value in the segment header. Instead, there are two fields called the sequence number and the acknowledgment number. These two fields refer to the byte number and not the segment number.

Byte Number

TCP numbers all data bytes that are transmitted in a connection. Numbering is independent direction. When TCP receives bytes of data from a process it stores them in the sending buffer and numbers them. The numbering does not necessarily start from 0. Instead, TCP generates a random number between 0 and 2^32-1 for the number of the first byte. For example, if the random number happens to be 1,057 and the total data to be sent is 6,000 bytes, the bytes are numbered from 1,057 to 7,056. We will see that byte numbering is used for flow and error control.

Sequence Number

After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment.

Example 2

Suppose a TCP connection is transferring a file of 5000 bytes. The first byte is numbered 10001. What are the sequence numbers for each segment if data is sent in five segments, each carrying 1000 bytes?

Solution

The following shows the sequence number for each segment:

Segment 1→Sequence number:10,001(range:10,001 to 11,000)

Segment 2→Sequence number:11,001(range:11,001 to 12,000)

Segment 3→Sequence number:12,001(range:12,001 to 13,000)

Segment 4→Sequence number:13,001(range:13,001 to 14,000)

Segment 5→Sequence number:13,001(range:14,001 to 15,000)

When a segment carries a combination of data and control information (piggybacking), it uses a sequence number. If a segment does not carry user data, it does not logically define a sequence number. The field is there, but the value is not valid. However, some segments, when carrying only control information need a sequence number to allow an acknowledgement from the receiver. These segments are used for connection establishment, termination, or abortion. Each of these segments consume one sequence number as though it carries one byte, but there is no actual data. If the random generated sequence number is x, the first data byte number is x+1. The byte x is considered as phony byte that is used for a control segment to open a connection, as we will see shortly.

Acknowledgement Number

As we discussed previously, communication in TCP is full duplex; when a connection is established, both parties can send and receive data at the same time. Each party numbers the bytes, usually with a different starting byte number. The sequence number in each direction shows the number of the first byte carried by the segment. Each party also uses an acknowledgement number to confirm the bytes it has received.

However, the acknowledgement number defines the number of the next byte that the party expects to receive. In addition, the acknowledgement number is cumulative here means that if a party uses 5,643 as an acknowledgement number, it has received all bytes from the beginning upto 5,642. Notr that this does not mean that the party has received 5,642 bytes because the first byte number does not have to start from 0.

Flow Control

TCP, unlike UDP, provides flow control. The receiver of the data controls how much data are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

Error Control

To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented, as we will see later.

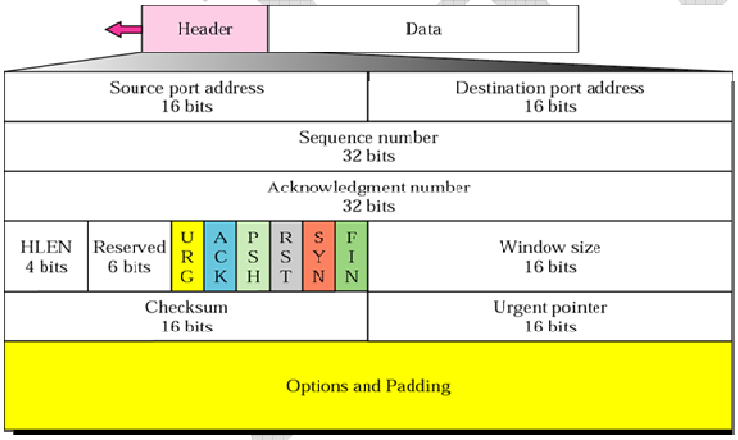
Congestion Control

TCP, unlike UDP, takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network.

SEGMENT

Before discussing TCP in more detail, let us discuss the TCP packets themselves. A packet in TCP is Calls a **segment.**

Figure 12.5 TCP segment format



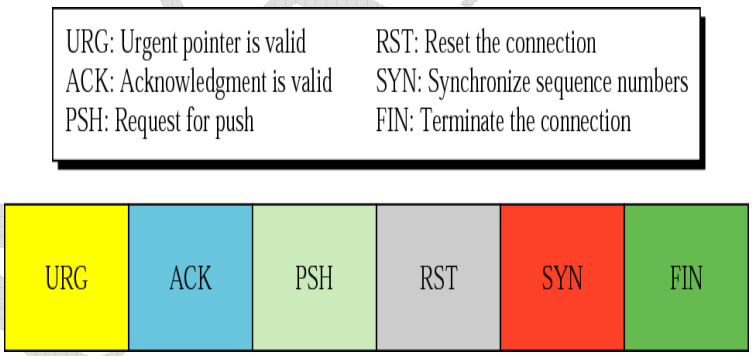
The segment consists of a 20- to 60-byte header, follwed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options. We will discuss some of the header fields in this section. The meaning and purpose of these will become clearer as we proceed through the chapter.

- **Source port address.** This is a 16-bit field that defines the port number of the application program in the host that is sending the segment. This serves the same purpose as the source port address in the UDP header discussed in Chapter 11.
- **Destination port address.** This is a 16-bit field that defines the port number of the application program in the host that is sending the segment. This serves the same purpose as the destination port address in the UDP header discussed in Chapter 11.

- **Sequence number.** This is a 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence compromises the first byte in the segment. During connection establishment (see Section 12.12) each party uses a random number generator to create an **initial sequence number** (ISN), which is usually different in each direction.
- **Acknowledgement number.** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received the byte number x from the other party, it defines x+1 as the acknowledgement number. Acknowledgement and data can be piggybacked together.

- **Header length.** This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 (5*4 = 20) and 15 (15 *4=60).
- **Reserved.** This is a 6-bit field reserved for future use.
- **Control**. This field defines 6 different control bits or flags as shown in Figure 12.6.One or more of these bits can be set at a time.

Figure 12.6 Control field.



These bits enable the flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP. A brief description of each bit is shown in table 12.2. We will discuss them further when we study the detailed operation of TCP later in the chapter.

Table 12.2 Description of flags in the control field.

Flag	*Description*
URG	The value of the urgent pointer field is valid
ACK	The value of the acknowledgment field is valid
PSH	Push the data
RST	The connection must be reset
SYN	Synchronize sequence numbers during connection
FIN	Terminate the connection

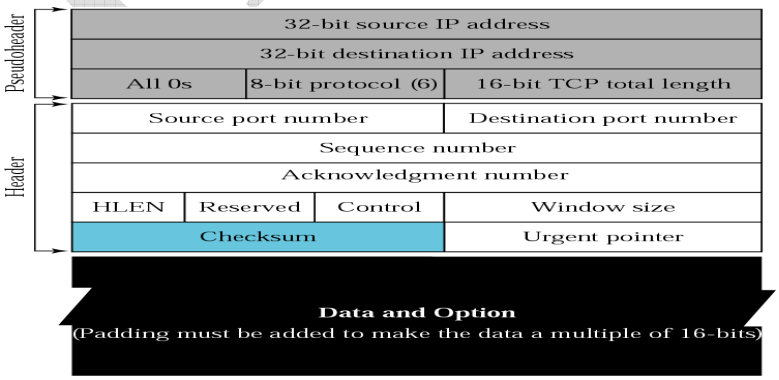
- **Window size.** This field defines the size of the window, in bytes, that the other party must maintain. Note that the length of this field is 16 bits, which means that the maximum size of window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.
- **Checksum.** This 16-bit field contain the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP in Chapter 11. However, the inclusion of the checksum in the UDP datagram is optional, whereas the inclusion of the checksum for TCP is mandatory. The same pseudoheader, serving the same purpose, is added to the segment. For the TCP pseudoheader, the value for the protocol field is six. See Figure 12.7.

Urgent pointer.

This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added the sequence number to obtain the number of the 1 ast urgent byte in the data section of the segment. This will be discussed later in this chapter.

- **Options.** There can be up to 40 bytes of optional information in the TCP header. We will discuss the different options currently used in the TCP header later in the chapter.

Figure Pseudoheader added to the TCP datagram



Encapsulation

A TCP segment is encapsulated in an IP datagram, which in turn is encapsulated in a frame at the data-link layer as shown in Figure 12.8

Figure Encapsulation and decapsulation

Transmission Control Protocol

TCP is a connection oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All of the segments belonging to a message are then sent over this virtual path. Using a single virtual pathway for the entire message facilities the acknowledgement process as well as retransmission of damaged or lost frames. You may wonder how TCP, which uses the services of IP, a connectionless protocol, can be connection-oriented. The point is that a TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. If a segment is lost or corrupted, it is retransmitted. Unlike TCP, IP is unaware of this retransmission. If a segment out of order, TCP holds it until the missing segments arrive; IP is unaware of this reordering. TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

Connection Establishment

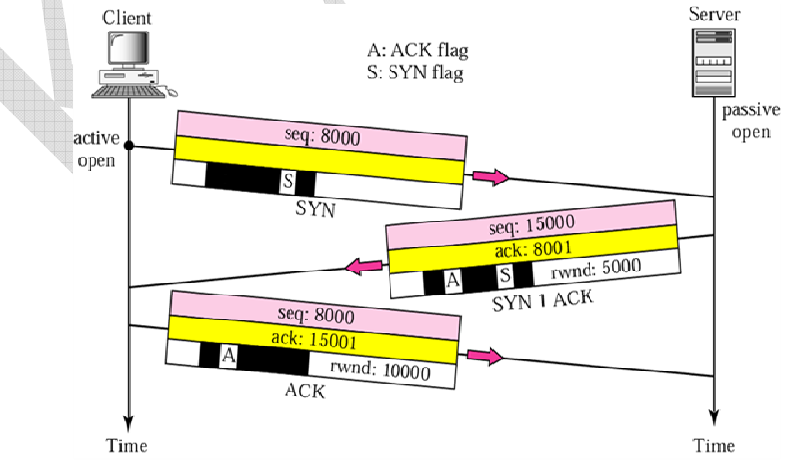
TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously. This implies that each party must initialize communication and get approval from the other party before any data is transferred.

The "three-way handshake" happens thus. The originator (you, hopefully) sends an initial packet called a "SYN" to establish communication and "synchronize" sequence numbers in counting bytes of data which will be exchanged. The destination then sends a "SYN/ACK" which again "synchronizes" his byte count with the originator and acknowledges the initial packet. The originator then returns an "ACK" which acknowledges the packet the destination just sent him. The connection is now "OPEN" and ongoing communication between the originator and the destination are permitted until one of them issues a "FIN" packet, or a "RST" packet, or the connection times out. All the protocols of the Internet which need "connections" are built on the TCP protocol. The "three way handshake" establishes the communication. Much like you picking up your phone, getting a dial tone, dialing the number, hearing ringing, and then the other party saying "hello" or "mushi mushi."

UDP is the other major underlying communication protocol of the Internet (besides TCP) - but it does not use a handshake to establish a "connection." It is much like a letter dropped in the mail. There

are no guarantees, and the post office makes a "best effort" to deliver the letter. But there is no final check to guarantee that the letter made it to its final destination. Of course there is a variant of the US mail which does this - registered mail. Operating behind a NAT/router has special challenges for UDP packets. A UDP IP packet just suddenly shows up at the router's doorstep, much like a letter, and it must try to decide whether to let it pass or not. Outgoing UDP packets are permitted pretty much without question, but incoming UDP packets (unsolicited UDP) are typically not permitted unless they correspond to previous UDP packets outgoing. The IP address is used to see if it is one that was addressed in a previous outgoing packet. There is then a timer placed on this; if a UDP packet is not seen on this connection for a certain time, then the router reverts to denying the incoming UDP. Now all the more familiar services of the Internet, such as web browsing (HTTP), email (POP3 and SMTP typically), FTP, Telnet, etc. etc. are built on top of the TCP and UDP protocols. And of course TCP and UDP are built on top of the fundamental IP protocol. IP protocols contain the IP addresses, TCP and UDP protocols contain the service addresses, to summarize it simply. IP packets are the fundamental "currency of the Internet. If you were to look under the covers of the Internet, you would see "IP packets" in the most general sense. At different places and different levels, you may see "ethernet frames" or "ATM cells" or "Frame Relay packets" - but these are just ways to carry the fundamental currency which is IP packets. The thing that makes the IP packet the fundamental currency of the Internet is that the IP packet contains two "Internet addresses," or "IP addresses."

Figure 12.9 *Connection establishment using three-way handshaking*



TCP/IP Protocol Suite 22

Connection Establishment Functions

The connection establishment process actually accomplishes several things as it creates a connection suitable for data exchange:

- **Contact and Communication:** The client and server make contact with each other and establish communication by sending each other messages. The server usually doesn't even know what client it will be talking to before this point, so it discovers this during connection establishment.
- **Sequence Number Synchronization:** Each device lets the other know what initial sequence number it wants to use for its first transmission.
- **Parameter Exchange:** Certain parameters that control the operation of the TCP connection are exchanged by the two devices.
- TCP Connection Establishment Sequence Number Synchronization and Parameter Exchange (Page 1 of 3)
- The TCP three-way handshake describes the mechanism of message exchange that allows a pair of TCP devices to move from a closed state to a ready-to-use, established connection. Connection establishment is about more than just passing messages between devices to establish communication.
- The TCP layers on the devices must also exchange information about the sequence numbers each device wants to use for its first data transmission, as well as parameters that will control how the connection operates.
- The former of these two data exchange functions is usually called sequence number synchronization, and is such an important part of connection establishment that the messages that each device sends to start the connection are called SYN (synchronization) messages.
- The Problem With Starting Every Connection Using the Same Sequence Number
- In the example I gave in the topic describing the sliding windows system, I assumed for "simplicity" (ha ha, was that simple?) that each device would start a connection by giving the first byte of data sent sequence number 1. A valid question is, why wouldn't we always just start off each TCP connection by sending the first byte of data with a sequence number of 1? The sequence numbers are arbitrary, after all, and this is the simplest method.
- In an ideal world, this would probably work, but we don't live in an ideal world. J The problem with starting off each connection with a sequence number of 1 is that it introduces the possibility of segments from different connections getting mixed up. Suppose we established a TCP connection and sent a segment containing bytes 1 through 30. However, there was a problem with the internetwork that caused this segment to be delayed, and eventually, the TCP connection itself to be terminated.

We then started up a new connection and again used a starting sequence number of 1. As soon as this new connection was started, however, the old segment with bytes labeled 1 to 30 showed up. The other device would erroneously think those bytes were part of the new connection.

- This is but one of several similar problems that can occur. To avoid them, each TCP device, at the time a connection is initiated, chooses a 32-bit initial sequence number (ISN) for the connection. Each device has its own ISN, and they will normally not be the same.
- Selecting the Initial Sequence Number
- Traditionally, each device chose the ISN by making use of a timed counter, like a clock of sorts, that was incremented every 4 microseconds. This counter was initialized when TCP started up and then its value increased by 1 every 4 microseconds until it reached the largest 32-bit value possible (4,294,967,295) at which point it "wrapped around" to 0 and resumed incrementing. Any time a new connection is set up, the ISN was taken from the current value of this timer. Since it takes over 4 hours to count from 0 to 4,294,967,295 at 4 microseconds per increment, this virtually assured that each connection will not conflict with any previous ones.
- One issue with this method is that it makes ISNs predictable. A malicious person could write code to analyze ISNs and then predict the ISN of a subsequent TCP connection based on the ISNs used in earlier ones. This represents a security risk, which has been exploited in the past (such as in the case of the famous Mitnick attack). To defeat this, implementations now use a random number in their ISN selection process.

TCP Connection Management and Problem Handling, the Connection Reset Function, and TCP "Keepalives" Once both of the devices in a TCP connection have completed connection setup and have entered the ESTABLISHED state, the TCP software is in its normal operating mode. Bytes of data will be packaged into segments for transmission using the mechanisms described in the section on message formatting and data transfer. The sliding windows scheme will be used to control segment size and to provide flow control, congestion handling and retransmissions as needed.

Once in this mode, both devices can remain there indefinitely. Some TCP connections can be very long-lived indeed—in fact, some users maintain certain connections like Telnet sessions for hours or even days at a time. There are two circumstances that can cause a connection to move out of the ESTABLISHED state:

Connection Termination: Either of the devices decides to terminate the connection. Connection Disruption: A problem of some sort occurs that causes the connection to be interrupted.

The TCP Reset Function

To allow TCP to live up to its job of being a reliable and robust protocol, it includes intelligence that allows it to detect and respond to various problems that can occur during an established connection. One of the most common is the half-open connection. This situation occurs when due to some sort of problem, one device closes or aborts the connection without the other one knowing about it. This means one device is in the ESTABLISHED state while the other may be in the CLOSED state (no connection) or some other transient state. This could happen if, for example, one device had a software crash and was restarted in the middle of a connection, or if some sort of glitch caused the states of the two devices to become unsynchronized.

To handle half-open connections and other problem situations, TCP includes a special reset function. A reset is a TCP segment that is sent with the RST flag set to one in its header. Generally speaking, a reset is generated whenever something happens that is "unexpected" by the TCP software. Some of the most common specific cases in which a reset is generated include: Receipt of any TCP segment from any device with which the device receiving the segment does not currently have a connection (other than a SYN requesting a new connection.) Receipt of a message with an invalid or incorrect Sequence Number or Acknowledgment Number field, indicating the message may belong to a prior connection or is spurious in some other way.

Receipt of a SYN message on a port where there is no process listening for connections.

Requirements and Issues in Connection Termination

Just as TCP follows an ordered sequence of operations to establish a connection, it includes a specific procedure for terminating a connection. As with connection establishment, each of the devices moves from one state to the next to terminate the connection. This process is more complicated than one might imagine it needs to be. In fact, an examination of the TCP finite state machine shows that there are more distinct states involved in shutting down a connection than in setting one up. The reason that connection termination is complex is that during normal operation, both of the devices are sending and receiving data simultaneously. Usually, connection termination begins with the process on just one device indicating to TCP that it wants to close the connection. The matching process on the other device may not be aware that its peer wants to end the connection at all. Several steps are required to ensure that the connection is shut down gracefully by both devices, and that no data is lost in the process.

Ultimately, shut down of a TCP connection requires that the application processes on both ends of the connection recognize that "the end is nigh" for the connection and stop sending data. For this reason, connection termination is implemented so that each device terminates its end of the connection separately.

The act of closing the connection by one device means that device will no longer send data, but can continue to receive it until the other device has decided to stop sending. This allows all data that is pending to be sent by both sides of the communication to be flushed before the connection is ended

2. User Datagram Protocol

User Datagram Protocol or UDP is part of the Internet Protocol suite, using which, programs running on different computers on a network can send short messages known as Datagrams to one another. UDP can be used in networks where TCP is traditionally implemented, but unlike TCP, it does not guarantee reliability or the correct sequencing of data. Datagrams may go missing without notice, or arrive in a different order from the one in which they were sent. The protocol was formulated by David P. Reed in 1980 and officially defined in RFC 768. UDP makes use of a simple communication model without implicit transmission checks for guaranteeing reliability, sequencing, or datagram integrity. Though these factors might seem to suggest that UDP is not a useful protocol, it still finds wide usage in particular areas where speed, more than reliability, is of utmost importance. UDP considers that error checks and corrections should be carried out in the communicating application, and not at the network layer. However, if error checks and corrections are needed at the network layer, the application can make use of Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP), which are specifically formulated for this reason. Since UDP does not have the overhead of checking whether the data has reached the destination every time it is sent, it makes the protocol that much faster and more efficient. UDP is often used for time-sensitive applications where missing data is preferred to late-arriving data. UDP is a stateless protocol which is useful for servers engaged in answering short queries from a large number of clients. While TCP is mainly used for communication between a server and a single client, UDP is used for packet broadcast or multicasting whereby the data is sent to all the clients in the network. Frequent network applications that use UDP include: Trivial File Transfer Protocol (TFTP) , Voice over IP (VoIP), IPTV, Domain Name System (DNS), etc. Since UDP lacks any kind of mechanism to control or avoid network congestion, other forms of network-based control mechanisms need to be implemented to ensure smooth flow of traffic in a UDP network. One of the solutions being designed to tackle this problem is DCCP or Datagram Congestion Control Protocol which is aimed at monitoring and controlling traffic congestion in a UDP network.

A typical IP network consists of five layers:

The Physical Layer consisting of the actual channel for data flow like coaxial, twisted pair or fiber optic cables The Data Link Layer implementing Wi-Fi, ISDN, GPRS etc The Network / Internet Layer Transport Layer implementing TCP, UDP etc Application Layer running DNS, FTP, HTTP, POP3,

SMTP, Telnet etc As shown above, UDP belongs to the fourth layer. Although the entire amount of UDP traffic in a network is a small fraction of the whole, a number of key application in the fifth layer like DNS and SNMP or simple network management protocol use UDP.

UDP Packet

The UDP header comprises of only four fields. The use of two of those is optional (light red background in diagram).

Bits	0 - 15	16 - 31
0	Source Port	Destination Port
32	Length	Checksum
64	Data	

Source Port

Source port recognizes the sending port and should be understood to be the port to respond to if required. If not used, then its value should be zero.

Destination Port

Destination port recognizes the destination port and is mandatory.

Length

A 16-bit length field indicates the length in bytes of the complete datagram: header and data.

Checksum

The 16-bit checksum field is implemented for error-checking of the header and data. The algorithm for computing the checksum is different for transmission over IPv4 and IPv6.

OSPF Protocol

Open Shortest Path First (OSPF) is a robust link-state interior gateway protocol (IGP). People use OSPF when they discover that Routing Information Protocol (RIP) just isn't going to work for their larger network, or when they need very fast convergence.

OSPF is the most widely used IGP. When we discuss IGPs, we're talking about one routing domain, or Autonomous System (AS). Imagine a medium-sized company with multiple buildings and departments, all connected and sharing two redundant Internet links. All of the buildings on-site are part of the same AS. With OSPF, however, we also have the concept of an Area, which allows further segmentation, perhaps by department in each building.

To understand the design needs for areas in OSPF, let's start with discussing how OSPF works. There is some terminology you may not have encountered before, including the following:

- Router ID: In OSPF this is a unique 32-bit number assigned to each router. This is chosen as the highest IP address on a router, and can be set large by configuring an address on a loopback interface of the chosen router.
- Neighbor Routers: two routers with a common link that can talk to each other.
- Adjacency: a two-way relationship between two neighbor routers. Neighbors don't always form adjacencies.
- LSA: Link State Advertisements are flooded; they describe routes within a given link.
- Hello Protocol: This is how routers on a network determine their neighbors and form LSAs.
- Area: a hierarchy. A set of routers that exchange LSAs, with others in the same area. Areas limit LSAs and encourage aggregate routes.
- OSPF is a link-state routing protocol, as we've said. Think of this as a distributed map of the network. To get this information distributed, OSPF does three things.

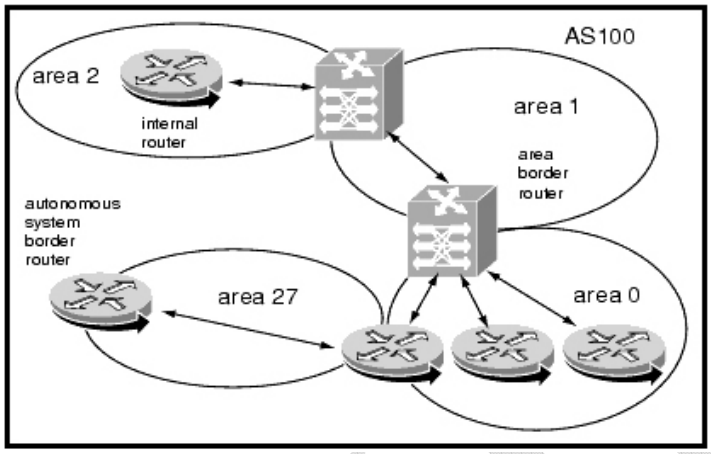
OSPF is a link-state routing protocol, as we've said. Think of this as a distributed map of the network. To get this information distributed, OSPF does three things.

First, when a router running OSPF comes up it will send hello packets to discover its neighbors and elect a designated router. The hello packet includes link-state information, as well as a list of neighbors. Providing information about your neighbor to that neighbor serves as an ACK, and proves that communication is bi-directional. OSPF is smart about the layer 2 topology: if you're on a point-to-point link, it knows that this is enough, and the link is considered "up." If you're on a broadcast link, the router must wait for an election before deciding if the link is operational.

The election ballot can be stuffed, with a Priority ID, so that you can ensure that your beefiest router is the Designated Router (DR). Otherwise, the largest IP address wins. The key idea with a DR and backup DR (BDR) is that they are the ones to generate LSAs, and they must do database exchanges with other routers in the subnet. So, non-designated routers form adjacencies with the DR. The whole DR/BDR design is used to keep the protocol scalable. The only way to ensure that all routers have the same information is to make them synchronize their databases. If you have 21 routers, and want to bring another one up, then you'd have to form 21 new adjacencies. If you centralize the database, with a backup (just in case), then adding more becomes an easy to manage linear problem.

The database exchange is part of bringing up adjacencies after the hello packets are exchanged, and it's very important. If the databases are out of sync, we could risk routing loops, blackholes and other perils. The third part of bringing up an adjacency is Reliable Flooding, or LSA exchange. The LSA area zero is special, and if you have multiple areas, they must all touch area zero. This is also called the

Backbone Area. There are different types of areas in OSPF, and it can get really crazy when you throw in Virtual Links to allow two areas to speak without hitting area zero.



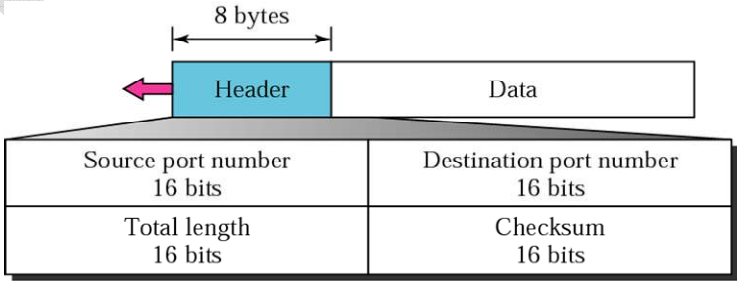
There also are different types of routers in OSPF:

- ABR: An Area Border Router is a router that is in area zero, and one or more other areas.
- DR, BDR: A Designated Router, as we said, is the router that keeps the database for the subnet. It sends and receives updates (via multicast) from the other routers in the same network.
- ASBR: The Autonomous System Boundary Router is very special, but confusing. The ASBR connects one or more AS, and exchanges routes between them. The ASBR's purpose is to redistribute routes from another AS into its own AS.

3. USER DATA GRAM

UDP packets, called **user datagrams**, have a fixed-size header of 8 bytes. Figure 11.7 shows the format of a user datagram.

Figure 11.7 *User datagram format*



- **Source port number.** This is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 t0 65,535. If the source host is the client (a client sending a request), the port number, in most cases, is an ephemeral port number requested by the process and chosen by the UDP software running on the source host. If the source host is the server (a server sending a response), the port number, in most cases, is a well-known port number.
- **Destination port number.** This is the port number used by the process running on the destination host. It is also 16 bits long. If the destination host is the server (a client sending a request), the port number, in most cases,is well-known port number. If the destination host is the client (a server sending a response), the port number, in most cases, is an ephemeral port number. In this case the server copies the ephemeral port number it has received in the request packet.

The fields are as follows:

- **Source port number.** This is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 t0 65,535. If the source host is the client (a client sending a request), the port number, in most cases, is an ephemeral port number requested by the process and chosen by the UDP software running on the source host. If the source host is the server (a server sending a response), the port number, in most cases, is a well-known port number.
- **Destination port number.** This is the port number used by the process running on the destination host. It is also 16 bits long. If the destination host is the server (a client sending a request), the port number, in most cases,is well-known port number. If the destination host is the client (a server sending a response), the port number, in most cases, is an ephemeral port number. In this case the server copies the ephemeral port number it has received in the request packet.
- **Length.** This is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be much less because an UDP user datagram is stored in an IP datagram with the total length of 65535 bytes.
- The length field in a UDP user datagram is actually not necessary. A user datagram is encapsulated in an IP datagram. There is a field in the IP datagram that defines the total length. There is another field in the IP datagram that defines the length of the header. So if we subtract the value of the second field from the first,we can deduce the length of the UDP datagram that is encapsulated in an IP datagram.

However, the designers of the UDP protocol felt that it was more efficient for the destination UDP to calculate the length of the data from the information provided in the UDP user datagram rather than asking the IP software to supply this information. We should remember that when the IP software delivers the UDP user datagram to the UDP layer, it has already dropped the IP header.

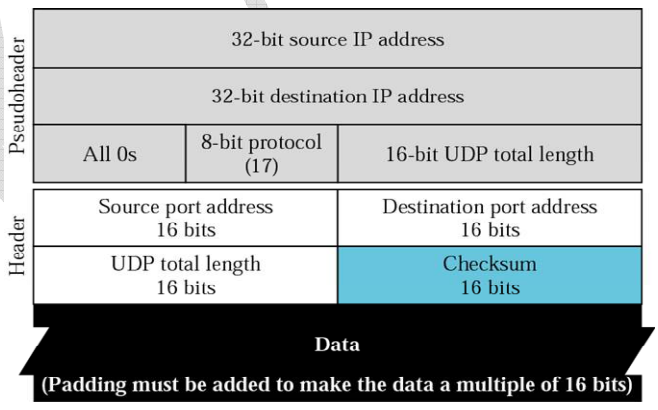
Checksum: This field is used to detect errors over the entire user datagram (header plus data). The checksum is discussed in the next section. We have already talked about the concept of the checksum and the way it is calculated. We have also shown how to calculate the checksum for the IP and ICMP packet; we now show how this is done for UDP.

UDP checksum calculation is different from the one for IP and ICMP. Here the checksum includes three sections: a pseudoheader, the UDP header, and the data coming from the application layer. The pseudoheader is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with 0s(see Figure 11.8).

UDP checksum calculation is different from the one for IP and ICMP. Here the checksum includes three sections:a pseudoheader, the UDP header, and the data coming from the application layer. The pseudoheader is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with 0s (see Figure 11.8).

UDP checksum calculation is different from the one for IP and ICMP. Here the checksum includes three sections a pseudoheader, the UDP header, and the data coming from the application layer.

Figure 11.8 *Pseudoheader for checksum calculation*



If the checksum does not include the pseudoheader, a user datagram may arrive safe and sound. However, if the IP address is corrupted, if may be delivered to the wrong host. The protocol field is added to ensure that the packet belongs to UDP, and not to TCP. We will see later that if a process can use either

UDP or TCP, the designation port number can be the same. The value of the protocol field for UDP is 17. if this value is changed during transmission, the checksum calculation at the receiver will detect it and UDP drops the packet. It is not delivered to the wrong protocol. Note the similarities between the pseudoheader fields and the last 12 bytes of the IP header.

Checksum calculation

1. **Checksum calculation at sender**
2. **Checksum calculation at Receiver**

Checksum calculation at sender

The sender follow this eight steps to calculate the checksum:

1. Add the pseudoheader to the UDP user datagram.
2. Fill the checksums fields with zeros.
3. Divide the total bits into 16-bit (2-byte) words.
4. If the total number of bytes even, add 1 byte of padding (all 0s). The padding is only for the purpose of calculating the checksum and will be discarded afterwards.
5. Add all 16-bit sections using one's complement arithmetic.
6. Complement the result (change all 0s to 1s and all 1s to 0s), which is 16-bit number, and insert in the checksum field.
7. Drop the pseudoheader and any added padding.
8. Deliver the UDP user datagram to the IP software for encapsulation.

Note that the order of the rows in pseudoheader doesnot make any difference in checksum calculation. Also, adding 0s doesnot change the result. For this reason, the software that calculates the checksum can easily add the whole IP header (20 bytes) to the UDP datagram, set the first bytes to zero, set the TTL fields to zero, replace the IP checksum to UDP length, and calculate the checksum. The result would be the same.

Checksum Calculation at Receiver

The receiver follows these six steps to calculate the checksum:

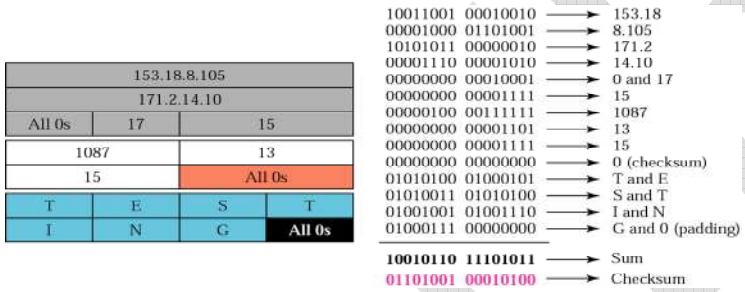
1. Add the pseudoheader to the UDP user datagram.
2. Add padding if needed.
3. Divide the total bits into 16-bit sections.
4. Add all 16-bit sections using one's complement arithmetic.
5. Complement the result.

6. If result is all 0s, drop the pseudoheader and any added padding and accept the user datagram. If the result is anything else, discard the user datagram.

An example

Figure 11.9 shows the checksum calculation for a very small user datagram with only 7 bytes of data. Because the number of bytes of data is odd, padding is added for checksum calculation. The pseudoheader as well as the padding will be dropped when the user datagram is delivered to IP.

Figure 11.9 *Checksum calculation of a simple UDP user datagram*



Optional use of the checksum

The calculation of the checksum and its inclusion in a user datagram is optional. If the checksum is not calculated, the field is filled with 0s. one might ask, when the UDP software on the destination computer receives a user datagram with a checksum value of zero, how can it determine if the checksum was not used or if it was used and the result happened to be all 0s? The answer is very simple. If the source does calculate the checksum and the result happens to be all 0s, it must be complemented. So what is sent is all 1s. Note that a calculated checksum can never be all 0s because this implies that the sum is all 1s which is impossible in two's complement arithmetic .

UDP Operation

UDP uses concept common to the transport layer. These concepts will be discussed here briefly, and then expanded in the next chapter on the TCP protocol.

Connectionless Services

As mentioned previously, UDP provides a **connectionless service.** This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagram's even if they are coming from the same source process and going to the same destination program. The user datagram's are not numbered. Also, there is no connection establishment and no

connection termination as is the case for TCP. This means that each user datagram can travel on a different path.

One of the ramifications of being connectionless is that the process that uses UDP cannot send a stream of data to UDP and expect UDP to chop them into different related user data grams. Instead each request must be small enough to fit into one user datagram. Only those processes sending short messages should use UDP.

Flow and Error Control

UDP is a very simple, unreliable transport protocol. There is no flow control, and hence no window mechanism. The receiver may overflow with incoming messages.

There is no error control mechanism in UDP except for the checksum. This means thus the sender does not know if message has been lost are duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded.

The lack of **flow control and error control** means that the process using UDP should provide for these mechanisms.

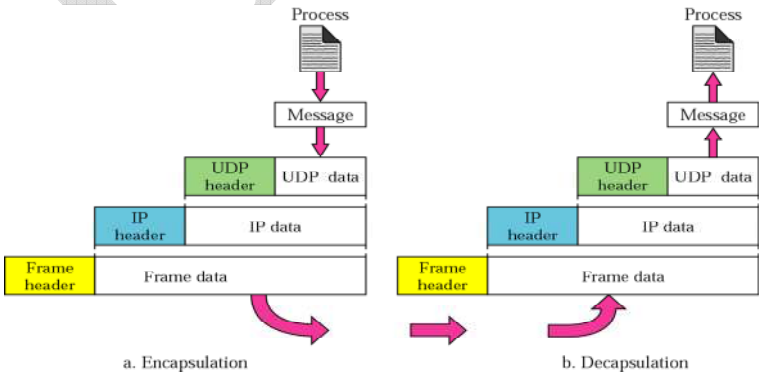
Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages (see figure 11.10).

Encapsulation

When a process has a message to send through UDP, it passes the message to UDP along with a pair of socket addresses and the length of data. UDP receives the data and adds the UDP header.

Figure 11.10 *Encapsulation and decapsulation*



UDP then passes the user datagram to IP with the socket addresses. IP adds its own header, using the value 17 in the protocol field,indicating that the data has come from the UDP protocol. The IP datagram is then passed to the data link layer. The data link layer receives the IP datagram, adds its own header(and possibly a trailer), and passes it to the physical layer. The physical layer encodes the bits into electrical or optical signals and sends it to the remote machine.

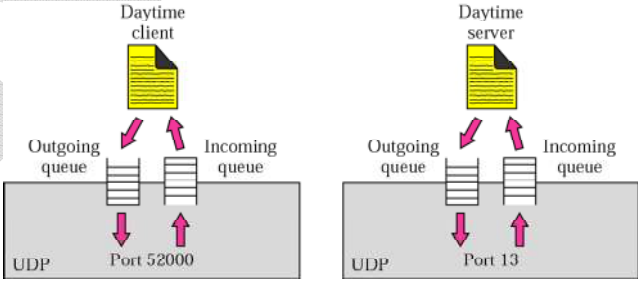
Decapsulation :

When the message arrives at the destination host, the physical layer decodes the signals into bits and passes it to the data link layer. The data link layer uses the header (and the trailer) to check the data. If there is no error, the header and trailer are dropped and the datagram is passed to **IP**. The IP software does its own checking. If there is no error, the header is dropped and the user datagram is passed to UDP with the sender and receiver IP addresses. UDP uses the checksum to check the entire user datagram. If there is no error, the header is dropped and the application data along with sender socket address is passed to the process. The sender socket address is passed to the process in case it needs to respond to the message received.

Queuing

We have talked about ports without discussing the actual implementation of them. In UDP queues are associated with ports (see figure 11.11). At the client site, when a process starts, it requests aport number from the operating systems. Some implementations create both an incoming and an outgoing queue associated with each process. Other implementations create only an incoming queue associated with each process.

Figure 11.11 ***Queues in UDP***



Note that even if a process wants to communicate with multiple processes, it obtains only one port number and eventually one outgoing and one incoming queue. The queues opened by the client

are, in most cases, identified by ephemeral port numbers. The queues function as long as the process is running. When the process terminates, the queues are destroyed.

The client process can send messages to the outgoing queue by using the source port number specified in the request. UDP removes the message one by one, and, after adding the UDP header, delivers them to IP. An outgoing queue can overflow. If this happens, the operating systems can ask the client process to wait before sending any more messages.

When a message arrives for a client, UDP checks to see if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue.

If there is no such queue, UDP discards the user datagram and asks the ICMP protocol to send a port unreachable message to be sent to the server. All of the incoming messages for one particular client program, whether coming from the same or different server, are sent to the same queue. An incoming queue can overflow. If this happens, UDP drops the user datagram and asks for a port unreachable message to be sent to the server.

At the server site, the mechanisms of creating queues are different. In its simplest form, a server asks for incoming and outgoing queues using its well-known port when it starts running. The queues remain open as long as the server is running.

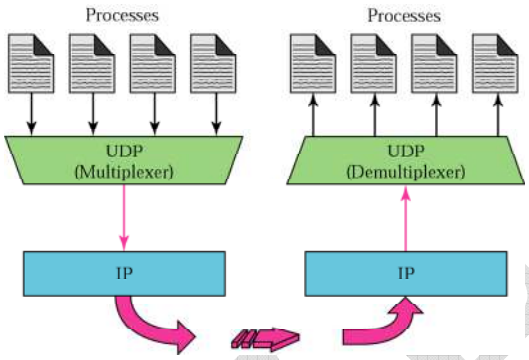
When a message arrives for a server, UDP checks to see if an incoming queue has been created for the port number specified in destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue. If there is no such a queue, UDP discards the user datagram and asks the ICMP protocol to send a port unreachable message to the client. All of the incoming messages for one particular server, whether coming from the same or a different client, are sent to the same queue. An incoming queue can overflow. If this happens, UDP drops the user datagram and asks for a port unreachable message to be sent to the client.

When a server wants to respond to a client, it sends messages to the outgoing queue using the source port number specified in the request. UDP removes the message one by one, and , after adding the UDP header, delivers them to IP. An outgoing queue can overflow. If this happens, the operating system asks the server to wait before sending any more messages.

Multiplexing and Demultiplexing

In a host running a TCP/IP protocol suite, there is only one UDP but possibly several processes that may want use the services of UDP. To handle this situation, UDP multiplexes and demultiplexes(see figure 11.12).

Figure 11.12 *Multiplexing and demultiplexing*



Multiplexing

At the sender site, there may be several processes that need to send user datagram's. However, there is only one UDP. This is a many-to-one relationship and requires multiplexing. UDP accepts messages from different processes, differentiated by their assigned port numbers. After adding the header, UDP passes the user datagram to the IP.

Demultiplexing

At the receiver site, there is only one UDP. However, we may have many processes that can receive user datagram's. This is a one- to-many relationship and requires demultiplexing. UDP receives user datagrams from IP. After error checking and dropping of the header, UDP delivers each message to the appropriate process based on the port numbers.

USES OF UDP

The following lists some of the uses of UDP protocol:

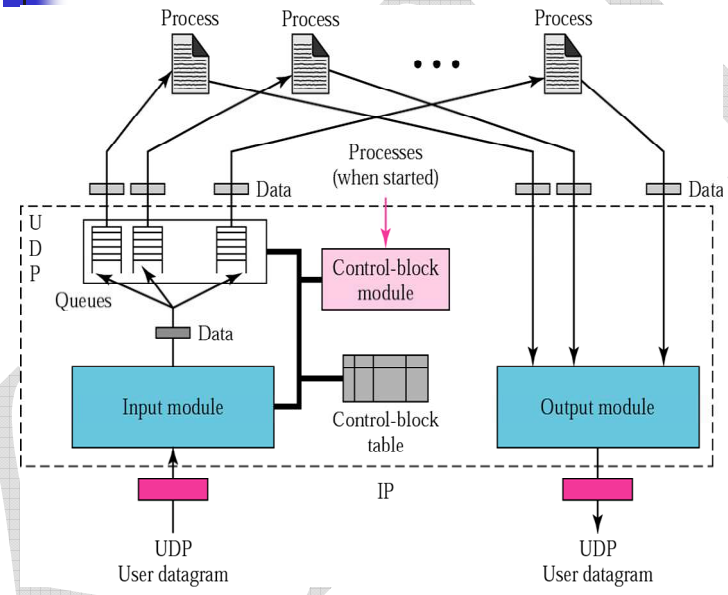
- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as FTP that needs to send bulk data
- UDP is suitable for a process with internal flow and error-control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) (see chapter 19) process includes flow and error control. It can easily use UDP.
- UDP is suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- UDP is used for management process such as SNMP

- UDP is used for some route updating protocols such as Routing Information Protocol (RIP) (see chapter 14).

UDP PACKAGE

- To show how UDP handles the sending and receiving of UDP packets, we present a simple version of the UDP package.
- We can say that the UDP package involves five components: a control-block table, input queues, a control-block module, an input module, and an output module. Figure 11.13 shows these five components and their interactions.

Figure 11.13 *UDP design*



Control-Block Table

In our package, UDP has a control-block table to keep track of the open ports. Each entry in this table has a minimum of four fields: the state, which can be FREE or IN-use, the process ID, the port number, and the corresponding queue number.

Input Queues

Our UDP packages use a set of input queues, one for each process. In this design, we do not use output queues.

Control-Block Module

The control-block module is responsible for the management of the control-block table. When a process starts, it asks for a port number from the operating system. The operating system assigns well-

known port numbers to servers and ephemeral port numbers to clients. The process passes the process ID and the port number to the control-block module to create an entry in the table for the process. The module does not create the queues; the field for queue number has a value of zero. Note that we have not included a strategy to deal with a table that is full.

Control-Block Module

Receive: a process ID and a port number

1. Search the control block table for a free entry.
    1. If (not found)
        1.Delete an entry using a predefined strategy.
    2. Create a new entry with the state IN-USE.
    3. Enter the process ID and the port number.
    2. Return.

Input Module

The input module receives a user datagram from the IP. It searches the control-block table to find an entry having the same port number as this user datagram. If the entry is found, the module uses the information in the entry to en-queue the data. If the entry is not found, it generates an ICMP message.

Input Module

Receive: a user datagram from IP

1. Look for the corresponding entry in the control-block table.
    1. If (found)
        1. Check the queue field to see if a queue is allocated.
            1. IF (no)

1. Allocate a queue.

            2. Enqueue the data in the corresponding queue.
    2. If (not found)
        1. Ask the ICMP module to send an “unreachable port” message.
        2. Discard the user datagram.
2. Return.

Output Module

The output module is responsible for creating and sending user datagrams.

Output Module

Receive: data and information from a process

1. Create a UDP user datagram.
2. Send a user datagram.
3. Return.

Examples

In this section we show some examples of how our package responds to input and output. The control-block table at the start of our examples is shown in Table 11.2

Table 11.2 *The control-block table at the beginning of examples*

State	Process ID	Port Number	Queue Number
IN-USE	2,345	52,010	34
IN-USE	3,422	52,011	
FREE			
IN-USE	4,652	52,012	38
FREE			

EXAMPLE 2

The first activity is the arrival of a user datagram with destination port number 52,012. The input module searches for this port number and finds it. Queue number 38 has been assigned to this port, which means that the port has been previously used. The input module sends the data to queue 38. The control-block table does not change.

Table 11.3 *Control-block table after Example 3*

State	Process ID	Port Number	Queue Number
IN-USE	2,345	52,010	34
IN-USE	3,422	52,011	
IN-USE	**4,978**	**52,014**	
IN-USE	4,652	52,012	38
FREE			

Possible Questions

Two Mark Questions

1. What is IGMP?
2. List the set of operations done by IGMP.
3. Define a process.
4. List the protocol used in Transport layer.
5. What is connection oriented service?

Six Mark Questions

1. Discuss about IGMP messages in detail
2. What is the need for Slide window? Discuss in detail.
3. What is IGMP? Explain its operations.
4. Explain about TCP Segmentation.
5. Write in general about User Datagram.
6. Discuss about TCP flow control.
7. What is the protocol applied for an IGMP host? Discuss about it in detail.
8. Write a detail note on TCP services.
9. Discuss UDP Service in general.

What is the need for TCP? Discuss in detail about its features.

Unit - III

S.No	Question	Choice1	Choice2	Choice3	Choice4	Answer
1	The domain name space is divided into _____ sections.	2	4	3	none	3
2	The DNS client alled a _____maps a name to a address or an address to a name	resolver	register	server	none	resolver
3	IN _____ resolution the client sends its request to a server	Iterative	recursive	non recursive	non iterative	recursive
4	In _____resolution the client may send its request to multiple servers	iterative	recursive	non recursive	non iterative	iterative
5	Two types of DNS messages are _____	questions and resources	Question and answers	queries and response	resources and records	queries and response
6	_____ is a method where by an answer to query is stored in memory	queuing	stacking	querying	caching	caching
7	DNS uses an __pointer for duplicated domain name information in its message	offset	inset	static	dynamic	offset
8	A_____ maps each address to a unique name	address space	domain space	offset pointer	name space	name space
9	_____ is a sequence of characters without structures	hierarchal name space	flat name space	address name space	domain name space	flat name space
10	In _____ name space the names are defined in a inverted tree structure	hierarchical	flat	address	domain	domain
11	_____ is a sting with the maximum of 63 characters	label	domain name	FQDN	none	label
12	The tree can have only _____levels	126	127	128	138	128
13	_____ is a sequence of labels separated by dots	domain name	FQDN	address space	PQDN	domain name
14	_____ is a domain name that contains the full name of a host	PQDN	FQDN	QQDN	AQDN	FQDN
15	A _____ is sub tree of the domain name space	zone	PQDN	domain	FQDN	PQDN
16	____ is used when the name to be resolved belongs to the same site as client	PQDN	FQDN	address space	none	FQDN
17	What a server is responsible for or has authority over is called a _____	segment	zone	domain	packed	zone
18	A _____is a server whose zone consists of the whole tree	primary server	secondary server	com server	root server	root server
19	The _____ define registered hosts according to their generic behavior	country domain	generic domain	inverse domain	net domain	generic domain
20	The _____section uses two character country abbreviations	country domain	generic domain	country domain	net domain	country domain
21	_____ domain is used to map an address to a name	inverse	generic	country	net	inverse
22	The new domains are added to the DNS by a _____	resolver	registrar	server	none	registrar
23	_____record is used by the client to get information from a server	question	answer	query	authoritative	question
24	Mapping a name to an address or an address to a name is called ____resolution	domain address	client address	name address	server address	name address
25	___ allows organizations to use the global internet for private and public communication.	VPN	DNS	FTP	SNMP	VPN
26	A common technique to encrypt and authenticate in VPN is _____ .	internet security	IP security	network security	web security	IP security
27	____ involves the encapsulation of an encrypted IP datagram in a second outer datagram.	Ipsec	VPN	tunneling	none	tunneling
28	Mobile IP is an enhanced version of _____	TCP	IP	ARP	FTP	IP
29	The protocol that binds IP address and physical address is _____.	RARP	ATMWAN	ATMRP	ATMARP	ATMARP
30	_____ is a cell- switched network that can be a highway for an IP datagram.	ATM	SNMP	DNS	DHCP	ATM
31	_____ router receives an IP datagram.	exit-point	middle-point	entering-point	ending-point	entering-point
32	_____ connection is established between two end points by the network provider.	PVC	TPA	SHA	VPN	PVC
33	In a _____ connection each time a router wants to make a connection	PVC	SVC	SHA	none	SVC
34	An ATM network can be divided into _____ subnetworks.	physical	dynamic	static	logical	logical
35	In mobile IP the host has its orginal address called _____ address.	physical	logical	home	care-of	home
36	In mobile IP the host has its temporary address is called _____ address.	home	care of	physical	logical	care of
37	In mobile IP the care of address is associated with the ____ network.	home	LAN	foreign	All the above	foreign
38	_____ is usually a router attached to the home network of the mobile host.	home agent	foreign agent	mobile agent	none	home agent
39	_____ is usually a router attached to the foreign network of the mobile host.	home agent	foreign agent	mobile agent	none	foreign agent

III

40	When the mobile host act as a foreign agent called a _____ care of addresss.	located	allocated	co-located	none	co-located
41	The first phase in mobile communication is _____ discovery.	agent	user	server	client	agent
42	Mobile IP uses the router solicitation packet of _____.	IMCP	MICP	MCIP	ICMP	ICMP
43	_____ messages are encapsulated in a UDP user datagram.	resolution	registration	extension	identification	registration
44	_____ network private internet and access to the global internet.	hybrid	VPN	private	public	hybrid
45	The actual mail transfer is done through _____.	SDA	TTA	MTA	both a & b	MTA
46	The protocol that defines the MTA client and server in the internet is called __	SNMP	MTP	FTP	SMTP	SMTP
47	SMTP defines _____ commands.	13	14	24	15	14
48	_____ is a TCP/IP client- server application for copying files	FTP	SMTP	SNMP	ICMP	FTP
49	FTP uses _____ well- known TCP ports.	3	2	4	5	2
50	In FTP port _____ is used for the control connection.	20	21	22	23	21
51	In FTP port _____ is used for data connection.	20	21	22	23	20
52	In FTP the client sents port number to the server using _____ command.	PASV	PORT	LIST	OPEN	PORT
53	In FTP the control connection is made between _____ processes.	control	data transfer	data connection	data store	control
54	In FTP the data connection is made between _____ processes.	control	data transfer	data connection	none	data transfer
55	_____ is a general purpose client-server program.	NSFNET	CSNET	TELNET	ARPANET	CSNET
56	When a user logs into a local time sharing system it is called _____.	remote login	local login	security login	none	local login
57	When a user wants to access an application program he performs _____.	remote login	local login	security login	global login	remote login
58	___ driver pretends the characters are coming from a terminal.	non terminal	pseudo termainal	remote terminal	local termainal	pseudo termainal
59	Control characters can be used to handle _____ server.	local	global	remote	none	remote
60	In the _____ mode the client sends one character at a time to the server.	character	line	point	both a & c	character
61	In the _____ mode the client sends one line at a time to the server.	character	line	point	both a & c	line
62	A ___is a group of connected, communicating devices such asComputers and Printers.	Internet	Bridge	Network	Network.	Bridge

SYLLABUS

Unit - IV: BOOTP – DHCP – Address discovery and Binding. DNS – Name Space – DNS in Internet – Resolution – Resource Records

1. BOOTP Protocol

BOOTP is not a dynamic configuration protocol. When a client requests its IP addresses, the BOOTP server consults a table that matches the physical addresses of the client with its IP addresses. This implies that the binding between the physical addresses and the IP address of the client already exists. The binding is predetermined. However, what if a host moves from one physical network to another? What if a host wants a temporary IP address? BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator. BOOTP is a static configuration protocol.

The **Dynamic Host Configuration Protocol (DHCP)** has been devised to provide static and dynamic address allocation that can be manual or automatic.

Static Address Allocation

In this capacity DHCP acts like BOOTP. It is backward compatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server. A DHCP server has a database that statically binds physical addresses to IP addresses.

Dynamic Address Allocation

DHCP has a second database with pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available IP addresses and assigns an IP address for a negotiable period of time.

When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database. The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (like a subscriber to a service provider). DHCP provides temporary IP address for a limited period of time.

The addresses assigned from the pool are temporary addresses. The DHCP server issues a lease for a specific period of time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the choice to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

Manual and Automatic Configuration

One major problem with BOOTP protocol is that the table mapping the IP addresses to physical adresses needs to be manually configured. This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes. DHCP, on the other hand, allows both manual and automatic configurations. Static addresses are created manually; dynamic addresses are created automatically.

Packet Format

To make DHCP backward compatible with BOOTP, the designers of DHCP have decided to use almost the same packet format. They have only added a 1-bit flag to the packet. However, to allow different interactions with the server, extra options have been added to the option field. Figure 16.6 shows the format of a DHCP message. The new fields are as follows:

- **Flag.** A 1-bit flag has been added to the packet (the first bit of the unused field) to let the client specify a forced broadcast reply (instead of unicast) from the server. If the reply were to be unicast to the client, the destination IP address of the IP packet is the address assigned to the client. Since the client does not know its IP address, it may discard the packet. However, if the IP datagram is broadcast, every host will receive and process the broadcast message.
- **Options.** Several options have been added to the list of options. One option, with value 53 for the tag subfield (see figure 16.5), is used to define the type of interaction between the client and server.(see table 16.2). Other options define parameters such as lease time and so on. The options field in DHCP can be up to 312 bytes.

Figure 16.6 ***DHCP packet***

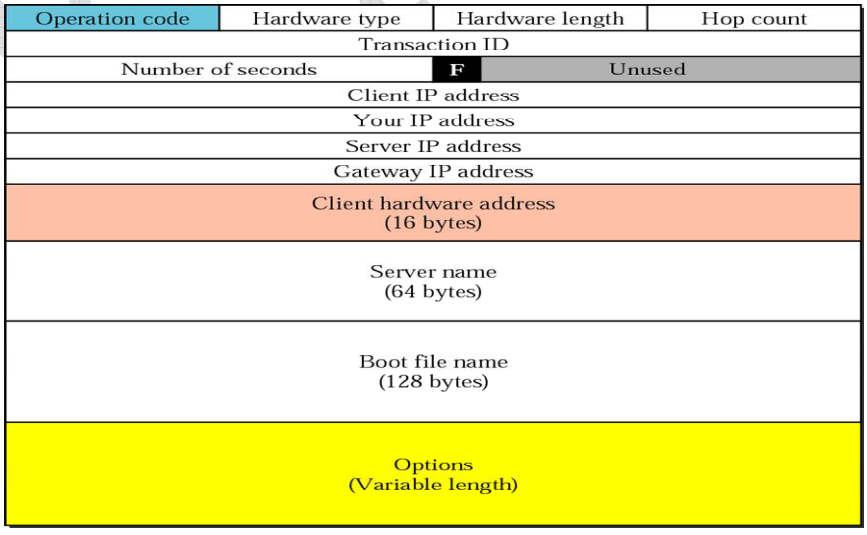


Table 16.2 Options for DHCP

Value	Value
1 DHCPDISCOVER	5 DHCPACK
2 DHCPOFFER	6 DHCPNACK
3 DHCPREQUEST	7 DHCPRELEASE
4 DHCPDECLINE	

We will see the use of these options in the next section.

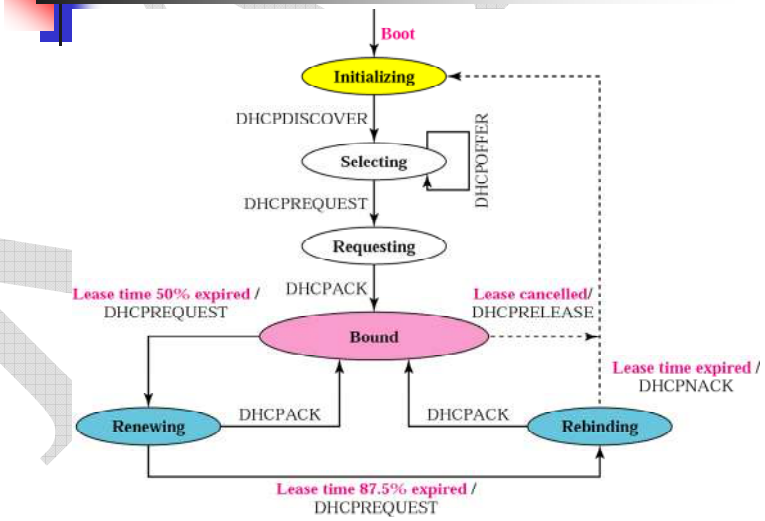
Transition states

The DHCP client transitions from one state to another depending on the messages it receives or sends. See figure 16.7.

Initializing state

When the DHCP client first starts, it is in the initializing state. The client broadcasts a DHCPDISCOVER message (a request message with the DHCPDISCOVER option) using port 67.

Figure 16.7 *DHCP transition diagram*



Selecting state

After sending the DHCPDISCOVER message, the client goes to the selecting state. Those servers that can provide this type of service respond with a DHCPOFFER message. In these messages, the servers offer an IP address. The can also offer the lease duration. The default is 1 h. The server that sends

a DHCPOFFER locks the offered IP address so that it is not available to any other clients. The client chooses one of the offers and sends a DHCPREQUEST message to be selected server. It then goes to the requesting state. However, if the client receives no DHCPOFFER message, it tries four more times, each with a span of 2s. If there is no reply to any of these DHCPDISCOVER, the client sleeps for 5 minutes before trying again.

Requesting state

The client remains in the requesting state until it receives a DHCPPACK message from the server which creates the binding between the client physical address and its IP address. After the receipt of the DHCPACK, the client goes to the bound state

Selectijtate

The client remains in the renewing state until one of two events happens. It can receive a DHCPACK, which renews the least agreement. In this case, the client resets and goes back to the bound state. Or, if a DHCPACK is not received, and 87.5% of the lease time expires; the client goes to the rebinding state.

Rebinding state

The client remains in the rebinding state until one of three events happens. If the client receives a CPNACK or the lease expires, it goes back to the initializing state and tries to get another IP address. If the client receives a DHCPACK it goes to the bound state and resets the timer.

Exchanging Messages

Figure 16.8 *Exchanging messages*

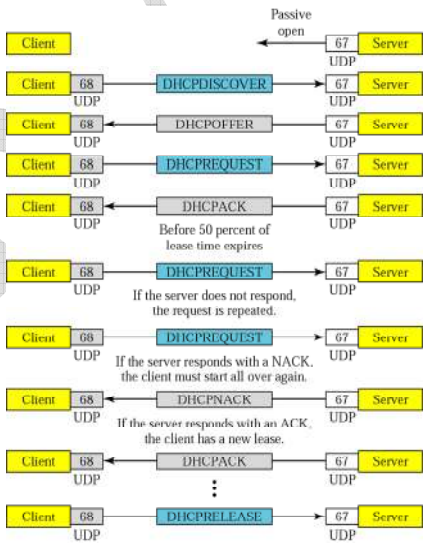


Figure 11.5 *ICANN ranges*

- **Well-known ports**. The ports ranging from 0 to 1,023 are assigned and controlled by ICANN. These are the well-known ports.
- **Registered ports.** The ports ranging from 1,024 to 1,023 are not assigned or controlled by ICANN. They can only be registered with ICANN to prevent duplication.
- **Dynamic ports**. The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used as temporary or private port numbers. The original recommendation was that the ephemeral port numbers for clients to be chosen from this range. However, most systems do not follow this recommendation.

Well-known Ports for UDP

Table 11.1 shows some well-known port numbers used by UDP. Some port numbers can be used by both UDP and TCP.

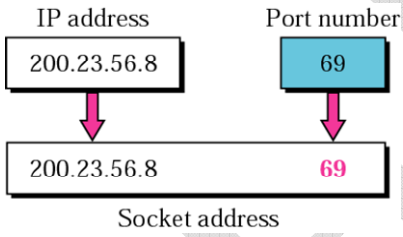
Table 11.1 Well-known ports used with UDP

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

Socket Addresses

As we have seen, UDP needs two identifiers, the IP address and the port number, at each end to make a connection. The combination of an IP address and a port number is called a **socket address**. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely (see figure 11.6).

Figure 11.6 *Socket address*



To use the services of UDP, we need a pair of socket addresses: the client socket address and the server socket address. These four pieces of information are part of the IP header and the UDP header. The ip header contains the IP addresses; the UDP header contains the port numbers.

2. DOMAIN NAME SYSTEM (DNS)

To identify an entity, TCP\IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name.

When the Internet was small, mapping was done using a *host file*. The host file had only two columns: name and address. Every host could store the host file on its disk and update it periodically from a master host file. When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.

Today, however, it is impossible to have one single host file to relate every address with a name and vice versa. The host file would be too large to store in every host. In addition, it would be impossible to update all the host files every time there is a change.

One solution would be to store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping, But we know that this would create a huge amount of traffic on the Internet.

Another solution , the one used today, is to divide this huge amount of information into smaller parts and store each part on a different computer. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS)

NAME SPACE

To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses. In other words, the names must be unique because the addresses are unique. A name space that maps each addresses to a unique name can be organized in two ways: flat or hierarchical.

Flat Name Space

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section: if they do, it has no meaning. The main disadvantages of a flat name space are that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication. Hierarchical Name Space

In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, and the third part can define departments in the organization, and do on. In this case, the authority to assign and control the name spaces can be decentralized. A central authority can assign the part of the name that defines the nature of the organization and the name of the organization. The responsibility of the rest of the name can be given to the organization itself. The organization can add suffixes (or prefixes) to the name to define its host or recourses. The management of the organization need not worry that the prefix chosen for a host is taken by another organization because, even if part of an address is the same, the whole address is different. For example, assume two colleges and a company call one of their computers challenger. The first college is given a name by the central authority such as fhda.edu; the second college is given the name smart.com. When each of these organizations adds the name challenger to the name they have already been given, the end result is three distinguishable names: challenger.fhda.edu, challenger.berkely.edu, and challenger.smart.com. The names are unique without the need for assignment by a central authority. The central authority controls only part of the name, not the whole.

DOMAIN NAME SPACE

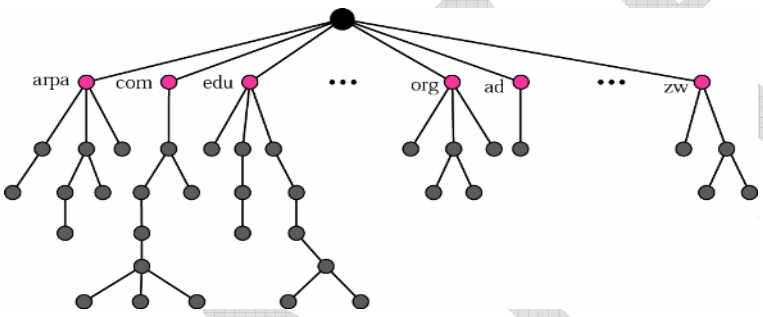
To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127 (see Figure).

Label

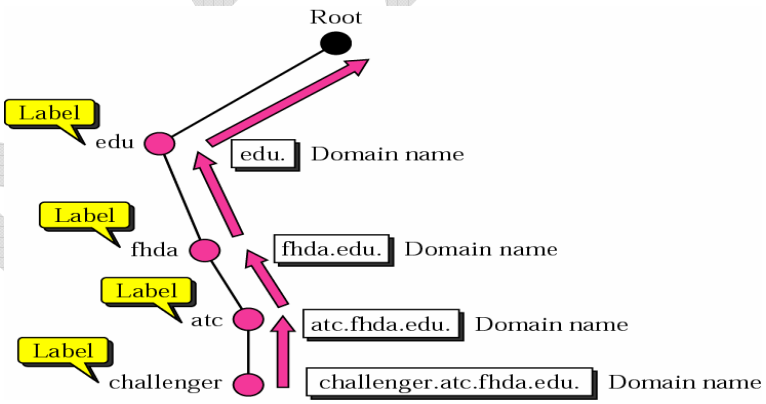
Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name

Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root.



The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.



Fully Qualified Domain Name (FQDN)

If a label is terminated by a null string, it is called a fully qualified domain name (FQDN). An FQDN is a domain name that contains the full name of a host. It contains all labels, from the most specific to the most general, that uniquely define the name of the host. For example, the domain name

challenger.atc.fhda.edu. is the FQDN of a computer named challenger installed at the Advanced Technology Center (ATC) at De Anza College. A DNS server can only match an FQDN to an address. Note that the name must end with a null label, but because null means nothing, the label ends with a dot (.).

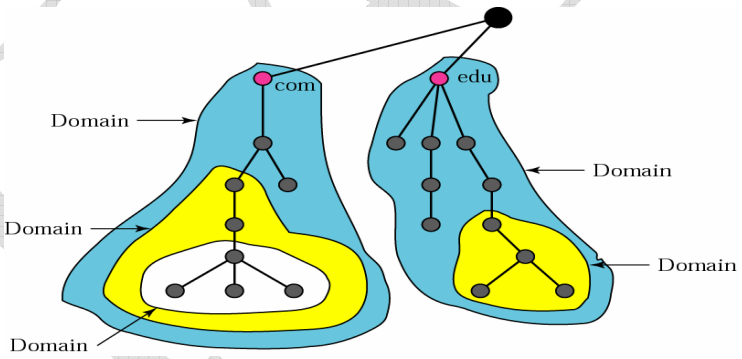
Partially Qualified Domain Name (PQDN)

If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN). A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client. Here the resolver can supply the missing part, called the suffix, to create an FQDN. For example, if a user at the fhda.edu. Site wants to get the IP address of the challenger computer, he or she can define the partial name

FQDN	PQDN
challenger.atc.fhda.edu.	challenger.atc.fhda.edu
cs.hmme.com.	cs.hmme
www.funny.int.	www

Domain

A domain is a sub-tree of the domain name space. The name of the domain is the domain name of the node at the top of the sub-tree. The above figure shows some domains. Note that a domain may itself be divided into domains (or subdomains as they are sometimes called).

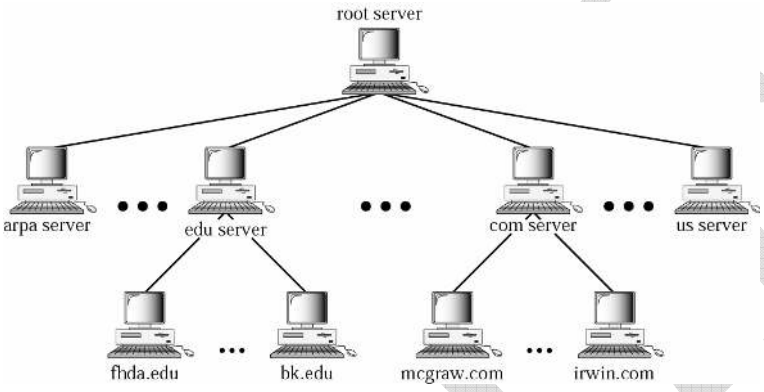


DISTRIBUTION OF NAME SPACE

The information contained in the domain space must be stored. However, it is very inefficient and also not reliable to have just one computer store such huge amount of information. It is inefficient because responding to request from all over the world places a heavy load on the system. It is not reliable because any failure makes the data inaccessible.

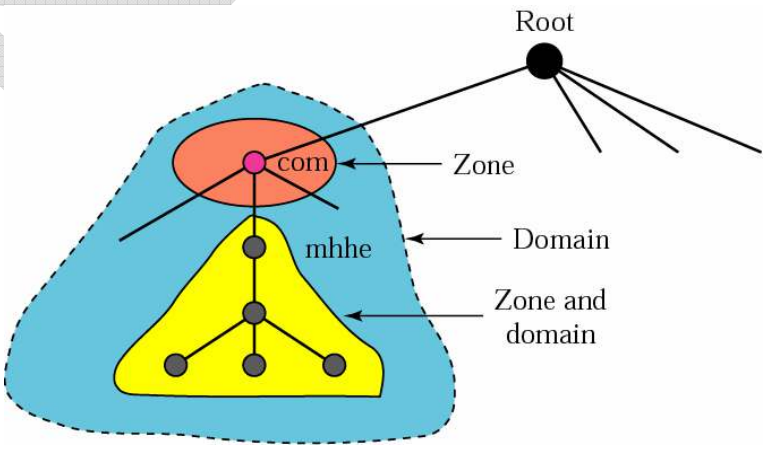
Hierarchy of Name Servers

The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level. In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes. Because a domain created this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or small domain. In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names (see below figure).



Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree. If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the "domain" and the "zone" refer to the same thing. The server makes a database called a zone file and keeps all the information for every node under that domain. However, if a server divides its domain into sub-domains and delegates part of its



Authority to others servers, "domain" and "zone" refer to different things. The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers. Of course the original server does not free itself from responsibility totally: It still has a zone, but the detailed information is kept by the lower-level servers.

A server can also divide part of its domain and delegate responsibility but still keep part of the domain for itself. In this case, its zone is made of detailed information for the part of the domain that is not delegated and references to those parts that are delegated.

Root Server

A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The servers are distributed all around the world.

Primary and Secondary Servers

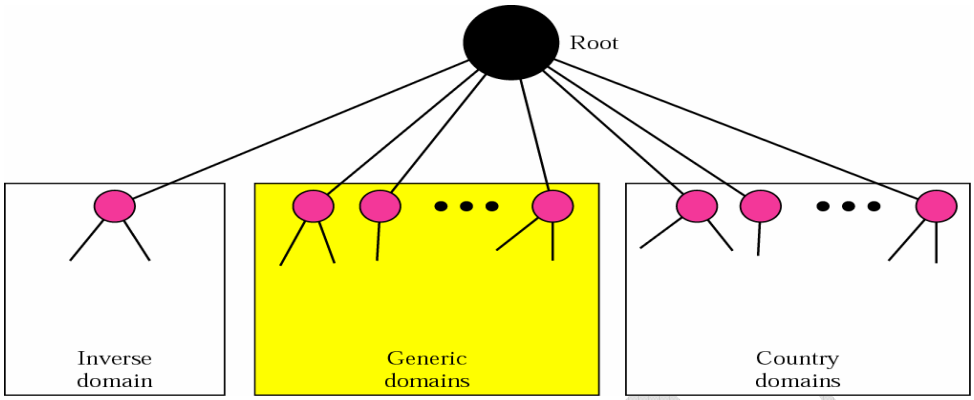
DNS defines two types of servers: primary and secondary. A primary server is a server that stores a file about the zone for which it is an authority. It is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.

A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary.

The primary and secondary servers are both authoritative for the zones they serve. The idea is not to put the secondary server at a lower level of authority but to create redundancy for the data so that if one server fails, the other can continue serving clients. Note also that a server can be a primary server for a specific zone and a secondary server for another zone. Therefore, when we refer to a server as a primary or secondary server, we should be careful to which zone we refer.

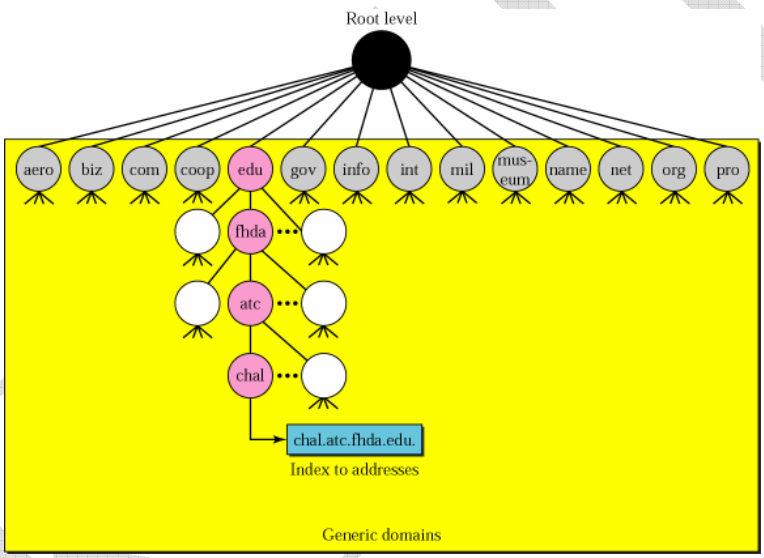
DNS in the internet

DNS is a protocol that can be used in different platforms. In the internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain.



Generic domains

The generic domains define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database.

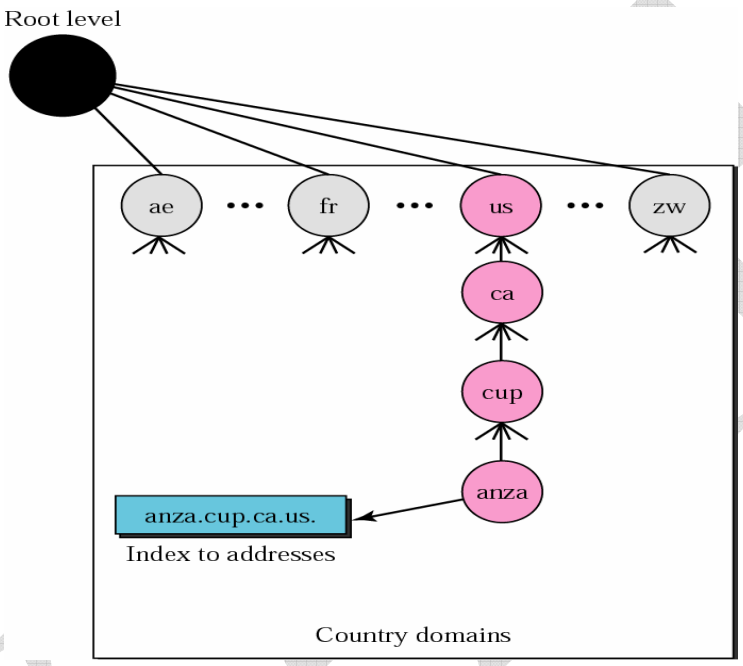


Looking at the tree, we see that the first level in the generic domains section allows 14 possible labels. These labels describe the organization types as listed in table.

Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers

Country Domains

The country domains section uses two-character country abbreviations (e.g., us for United States). Second-labels can be organizational, or they can be more specific, national designations. The United States, for example, uses state abbreviations as a subdivision of us (e.g.,ca.us.). The below figure shoes the country domains section. The address anza.cup.ca.us can be translated to De Anza College in Cupertino in California in the United States.



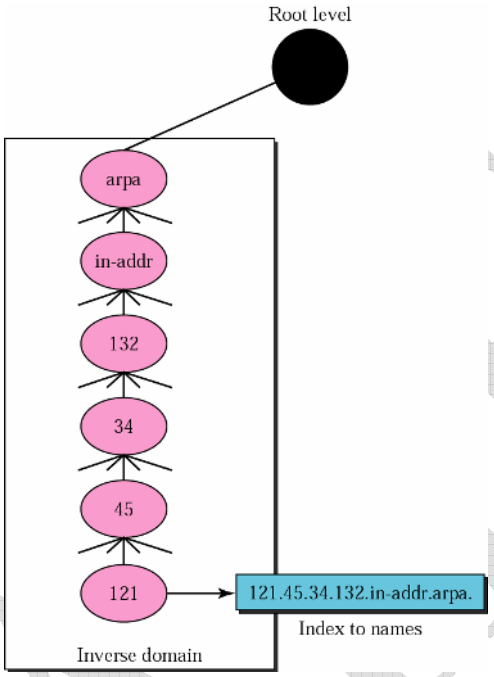
Inverse domain

The inverse domain is used to map as address to a name. This may happen, for example, when a server has received a request from a client to do a task. Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IPpacket) is listed. The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list.

This type of query is called an inverse or pointer (PTR) query.To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called arpa (for historical reason). The second level is also one single node named in-addr(for inverse address). The rest of the domain defines IP addresses.

The servers that handle the inverse domain are also hierarchical. This means the netid part of the address should be at a higher level than the subnetid part, and the subnetid part higher than the hostid part. In this way, a server serving the whole site is at a higher level than the servers serving each subnet. This

configuration makes the domain look inverted when compared to a generic or country domain. To follow the convention of reading the domain labels from the bottom to the top, an IP address such as 132.34.45.121(a classs B address with netid132.34) is read as 121.45.34.132.in-addr.arpa.



Registrar

How are the new domains added to DNS? This is done through a registrar, a commercial entity accredited by ICANN. A registrar first verifies that the requested domain name is unique and then enters into the DNS database. A fee is charged.

Resolution

Mapping a name to an address or an address to a name is called name-address resolution.

Resolver

DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver. The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information. After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.

Mapping Names to Addresses

Most of the time, the resolver gives a domain name to the server and asks for the corresponding address. In this case, the server checks the generic domains or the country domains to find the mapping.

If the domain name is from the generic domains section, the resolver receives a domain name such as "chal.atc.fhda.edu.". The query is sent by the resolver to the local DNS server for resolution. If the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly.

If the domain name is from the country domains section, the resolver receives a domain name such as "ch.fhda.cu.ca.us." The procedure is the same.

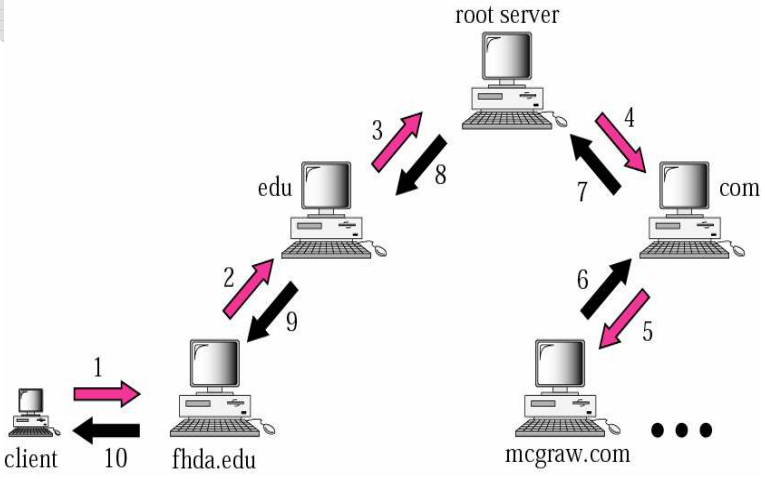
Mapping Addresses to Names

A client can send an IP address to a server to be mapped to a domain name. As mentioned before, this is called a PTR query. To answer queries of this kind, DNS uses the inverse domain. However, in the request, the IP address is reversed and two labels, in-addr and arpa, are appended to create a domain acceptable by the inverse domain section.

For example, if the resolver receives the IP address 132.34.45.121, the resolver first inverts the address and then adds the two labels before sending. The domain name sent is"121.45.34.132.in-addr.arpa.", which is received by the local DNS and resolved.

Recursive Resolution

The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer. If the server is the authority for the domain name, it checks its database and responds. If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response. If the parent is the authority, it responds; otherwise, it sends the query to yet another server. When the query is finally resolved, the response travels back until it finally reaches the requesting client.



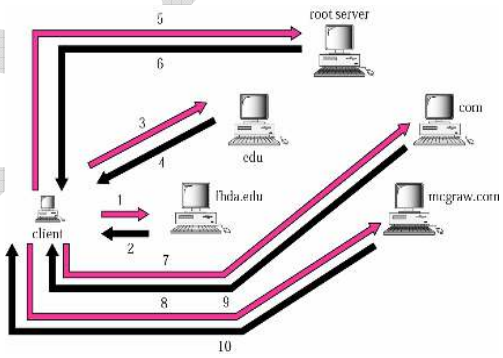
Iterative Resolution

If the client does not ask for a recursive answer, the mapping can be done iteratively. If the server is an authority for the name, it sends the answer. If it is not, it returns (to the client) the IP addresses of the server that it thinks can resolve the query. The client is responsible for repeating the query to this second server. If the newly addressed server can resolve the problem, it answers the query with the IP address; otherwise it the IP address of a new server to the client. Now the client must repeat the query to the third server. This process is called iterative because the client repeats the same query to multiple servers.

Caching

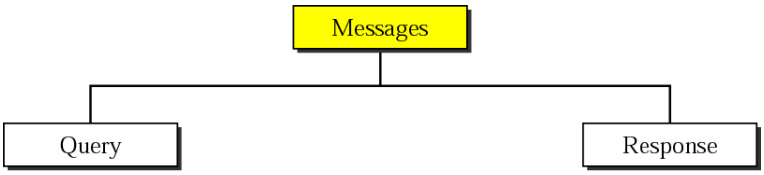
Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address. Reduction of this search time would increase efficiency. DNS handles this with a mechanism called caching. When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client. If the same or another client asks for the same mapping, it can check its cache memory and resolve the problem. However, to inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as un-authoritative.

Caching speeds up resolution, but it can also be problematic. If a server caches a mapping for a long time, it may send an outdated mapping to the client. To counter this two techniques are used. Fist, the authoritative server always adds information to the mapping called time-to-live(TTL). It defines the time in seconds that the receiving server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server. Second, DNS requires that each server keep a TTL counter for each mapping it caches. The cache memory must be searched periodically and those mappings with an expired TTL must be purged.

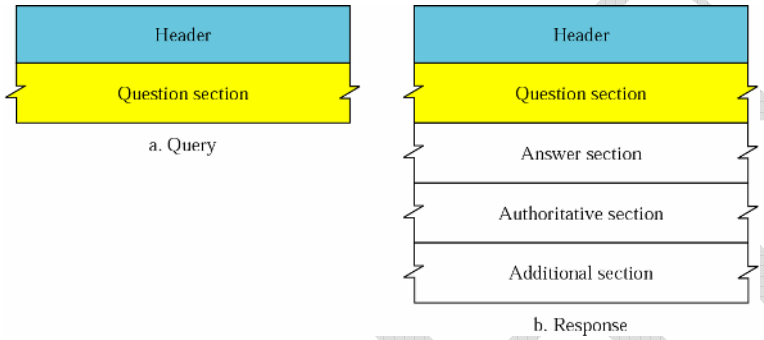


DNS MESSAGES

DNS has two types of messages: query and response (see below figure). Both types have the same format



The query message consists of a header and question records; the response message consists of a header, question records, answer records, authoritative records, and additional records (see below figure).



Header

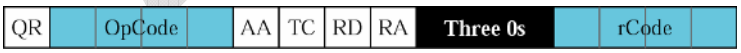
Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes and its format is shown in below figure.

Identification	Flags
Number of question records	Number of answer records (All 0s in query message)
Number of authoritative records (All 0s in query message)	Number of additional records (All 0s in query message)

The header fields are as follows:

Identification. This is a 16-bit field used by the client to match the response with the query. The client uses a different identification number each time it sends a query. The server duplicates this number in the corresponding response.

Flags. This is a 16-bit field consisting of the subfields shown in the below figure.



A brief description of each flag subfield follows.

- QR (query/response). This is a 1-bit subfield that defines the type of message. If it is 0, the message is a query. If it is 1, the message is a response.
- OpCode. This is a 4-bit subfield that defines the type of query or response (0 if standard, 1 if inverse, and 2 if a server status request).

- AA (authoritative answer). This is a 1-bit subfield. When it is set (value of 1) it means that the name server is an authoritative server. It is used only in a response message.
- TC (truncated). This is a 1-bit subfield. When it is set (value of 1), it means that the response was more than 512 bytes and truncated to 512. it is used when DNS uses the services of UDP.
- RD (recursion desired). This is a 1-bit subfield. When it is set (value of 1) it means the client desires a recursive answer. It is set in the query message and repeated in the response message.
- RA (recursion available).This is a 1-bit subfield. When it is set in the response, it means that a recursive response is available. It is set only in the response message.
- Reserved. This is a 3-bit subfield set to 000.
- rCode. This is a 4-bit field that shows the status of the error in the response. Of course, only an authoritative server can make a judgment. Table below shows the possible values for this field.

Value	*Meaning*
0	No error
1	Format error
2	Problem at name server
3	Domain reference problem
4	Query type not supported
5	Administratively prohibited
6–15	Reserved

- **Number of question records.** This is 16-bit field containing the number of queries in the section of the message.
- **Number of answer records.** This is 16-bit field containing the number of answer records in the answer section of the message. Its value is zero in the query message.
- **Number of authoritative records.** This is a 16-bit field containing the number of authoritative records in the authoritative section of a response message. Its value is zero in the query message.
- Number of additional records. This is a 16-bit field containing the number of additional records in the additional section of a response message. Its value is zero in the query message.

Question Section

This is a section consisting of one or more question records. It is present on both query and response message. We will discuss the question records in a following section.

Answer Section

This is a section consisting of one or more resource records. It is a present only on response messages. This section includes the answer from the server to the client (resolver). We will discuss resource records in a following section.

Authoritative Section

This is a section consisting of one or more resource records. It is present only on response messages. This section gives information (domain name) about one or more authoritative servers for the query.

Additional Information Section

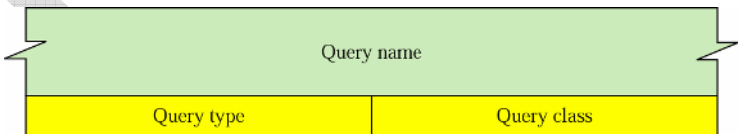
This is a section consisting of one or more resource records. It is present only on response messages. This section provides additional information that may help the resolver. For example, a server may give the domain name of an Authoritative server to the resolver in the authoritative section, and include the IP address of the same authoritative server in the additional information section.

TYPES OF RECORDS

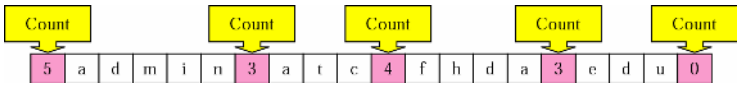
As we saw in the previous section, two types of records are used in DNS. The question records are used in the question section of the query and response messages. The resource records are used in the answer, authoritative and additional information sections of the response message.

Question Record

A question record is used by the client to get information from a server. This contains the domain name. The below figure shows a format of a question record. The list below describes question record fields.



Query name. This is a variable-length field containing a domain name (see below figure).



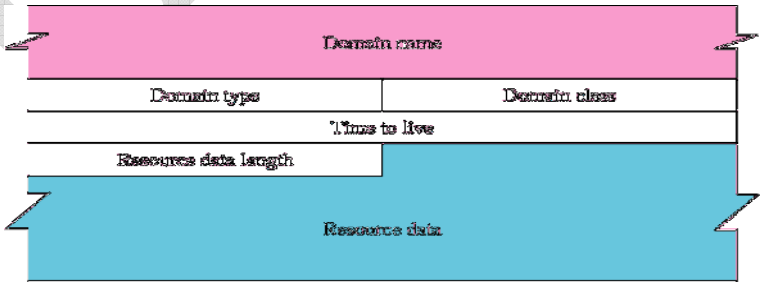
Query type. This is a 16-bit field defining the type of query. Table below shows some of the types commonly used. The last two can only be used in a query.

Type	*Mnemonic*	*Description*
1	A	Address. A 32-bit IPv4 address. It is used to convert a domain name to an IPv4 address.
2	NS	Name server. It identifies the authoritative servers for a zone.
5	CNAME	Canonical name. It defines an alias for the official name of a host.
6	SOA	Start of authority. It marks the beginning of a zone. It is usually the first record in a zone file.
11	WKS	Well-known services. It defines the network services that a host provides.
12	PTR	Pointer. It is used to convert an IP address to a domain name.
13	HINFO	Host information. It gives the description of the hardware and the operating system used by a host.
15	MX	Mail exchange. It redirects mail to a mail server.
28	AAAA	Address. An IPv6 address (see Chapter 27).
252	AXFR	A request for the transfer of the entire zone.
255	ANY	A request for all records.

Resource Record

Each domain name (each node on the tree) is associated with a record called the resource record. The server database consists of resource records. Resource records are also what is returned by the server to the client. The below figure shows the format of a resource record.

- **Domain name.** This is a variable-length field containing the domain name. It is a duplicate of the domain name in the question record. Since DNS requires the use of compression everywhere a name is repeated, this field is a pointer offset to the corresponding domain name field in the question record.
- **Domain type.** This field is the same as the query type field in the question record except the last two types are not allowed.



- **Domain class.** This field is the same as the query class field in the question record.

- **Time to live.** This is a 32-bit field that defines the number of seconds the answer is valid. The receiver can cache the answer for this period of time. A zero value means that the resource record is used only in a single transaction and is not cached.
- **Resource data length.** This is a 16-bit field defining the length of the resource data.
- **Resource data.** This is a variable-length field containing the answer to the query (in the answer section) or the domain name of the authoritative server (in the authoritative section) or additional information (in the additional information section). The format and contents of this field depend on the value of the type field. It can be one of the following:

a. **A number.** This is written in octets. For example, an IPv4 address is a 4-octet integer and an IPv6 address is a 16-octet integer.

A domain name. Domain names are expressed as a sequence of labels. Each label is preceded by a 1-byte length field that defines the number of characters in the label. Since every domain name ends with the null label, the last byte of every domain name is the length field with the value 0. To distinguish between a length field and an offset pointer (as we will discuss later), the two high-order bits of a length field always zero (00). This will not create a Problem because the length of a label cannot be more than 63, which is a maximum of 6 bits (111111).

- **An offset pointer.** Domain names can be replaced with an offset pointer. An offset pointer is a 2-byte fields with each of the 2 high-order bits set to 1 (11).
- **A character string.** A character string is represented by a 1-byte length field followed by the number of characters defined in the length field. The length field is not restricted like the domain name length field. The character string can be as long as 255 characters (including the length field).

Possible Questions

Two Mark Questions

1. What is address mapping?
2. What is the purpose of DSN?
3. What is a resource in DSN?
4. Define an URL.
5. What is DHCP?

Six Mark Questions

1. Write a general note on DHCP.
2. What are the types of configuration in DHCP? Discuss about it in detail.
3. Explain about DHCP operations on same and different network.
4. What is BOOTP? Discuss about it in detail
5. What is the need for DNS? Discuss about it in detail.
6. Discuss about name address Resolution of DNS in detail
7. Write a detailed note on DNS Namespace.
8. What is a Record in DNS? Explain its type with its format
9. Discuss in detail about DNS in Internet.
10. Discuss about Query and Response message type of DNS in detail

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed University Established Under Section 3 of UGC Act, 1956)
Coimbatore – 641021, INDIA
Department of Computer Science, Applications & Information Technology

Unit - IV

S.No	Question	Choice1	Choice2	Choice3	Choice4	Answer
1	Repeaters and Hub Operate in _____ Layer of Internet Model	First	Second	Third	First & Second.	First
2	Repeater receives _____ and regenerates _____ pattern.	Data, bytes	Signal, Bit	Decimal, Signal	None.	Signal, Bit
3	Repeater connects _____ of a LAN.	Part	Mode	Mode2	Segment	Segment
4	Router connects _____ Lan to create _____	Segment, Network	Sector, Internet	Independent, Internetwork	None.	Independent, Internetwork
5	Router changes physical address into _____	Datagram	Packet	Signal	bytes	Packet
6	_____notation uses Dot for separating bytes	Decimal	Binary	Hexa	Octal	Decimal
7	Migration is very fast in _____ addressing	Classful addressing	Classless addressing	Both	None	Classful addressing
8	If all the bits are one then it is a _____ address	Class A	Class B	Class C	Class E	Class E
9	A Block of address _____ is same.	First	Second	Third	Fourth.	First
10	The First address in a block is _____	Net-id	Host – id	Subnet	Network address	Network address
11	NIC stands for _____	Network Information Card	Network Information Center	Network Interface Card	Network Interface Center	Network Interface Card
12	_____ associates a logical address with physical address.	Static mapping	Dynamic mapping	Temporary mapping	Logical mapping	Static mapping
13	_____ addresses in the TCP/IP protocol suite are called IP address.	physical	static	dynamic	logical	logical
14	_____ defines the length of physical address in bytes.	protocol length	hardware length	software length	none	hardware length
15	_____ define the length of logical address in bytes	protocol length	hardware length	software length	none	protocol length
16	In cache table, _____ state means that the entry is complete.	Free	pending	resolved	none	resolved
17	_____ module is responsible for maintain the cache table.	Input module	cache control module	output module	queue	cache control module
18	TOS bits means	Type Of Service	Type Of Security	Type Of System	none	Type Of Service
19	_____ option used for padding at the end of the option field	end of option	checksum	operation option	no operation option	end of option
20	SCMP messages are divided into _____ broad categories	2	4	1	3	2
21	Destination unreachable message is _____ type of message.	Query	error reporting	query reporting	none	error reporting
22	address mask requesting or reply is _____ type of message.	error reporting	query	delay reporting	none	query
23	In destination _____represent the host is unreachable.	code1	code 2	code 3	code 4	code1
24	BOOTP stands for	Bootstrap protocol	Bootstrap project	Booting protocol	Booting project	Bootstrap protocol
25	_____ command that can create series of echo request and echo reply.	ping	pong	a and b	none	ping
26	_____ program in units can be used to trace the route of a packet	Tracer	Trace Route	Trace up	Trace Down	Trace Route
27	_____ programs in windows can be used to trace the route of a packet	Tracer	Trace Route	Trace up	Trace Down	Tracer
28	_____ msg in ICMP was designed to add a kind of flow control to the IP.	Source-Quench	Time-exceed	parameter problem	Re-direction	Source-Quench
29	In destination unreachable error reporting msg _____represent a protocol is unreachable.	code1	code 2	code 3	code 4	code 2
30	In service type TOS bit 0001 represent _____	Normal	Minimize Cost	Maximum reliability	Minimize delay	Minimize Cost
31	_____ is not a multicasting routing protocol.	ICMP	IGMP	TCP	TCP/IP	IGMP
32	IGMP stands for	Internet Group Management Protocol	Internet Group Maintenance Protocol	Information Group Management Protocol	Information Group Maintenance Protocol	Internet Group Management Protocol
33	_____ managers group membership	ICMP	IGMP	TCP	TCP/IP	IGMP
34	_____ is called a connectionless, unreliable transport protocol.	UDP	TCP	SCTP	FTP	UDP
35	UDP means _____	User Datagram Protocol	User Defined Protocol	User Derived Protocol	All the above	User Datagram Protocol
36	In Multicasting _____process multicast packet are encapsulated network.	Tunneling	Trimming	Transporting	none	Tunneling
37	UDP provides _____ service.	connection-oriented	connectionless	a and b	none	connectionless
38	_____ protocol provide domain name services	Echo	daytime	name server	quote	name server
39	_____ protocol returns string of characters.	daytime	quote	chargen	RPC	daytime
40	RPC means	Remote Procedure Call	Remote Packet Call	Resource Procedure Call	Response Procedure Call	Remote Procedure Call
41	The connection establishment in ____is called three way handshaking.	UDP	FTP	TCP	TCP/IP	FTP
42	ISN means _____.	Initial Sequence Number	Initial Service Number	Initial Segment Number	Initial Segment Node.	Initial Sequence Number
43	_____ is a string of characters that hold some information	country domain	compression	cookie	none	cookie
44	In TCP one end can store sending data while still receiving data is called.	full-close	half close	two-way handshaking	three way handshaking	half close
45	A _____ is a machine that goes through a limited no of rates.	Infinite state machine	finite state machine	unlimited statemachine	limited state machine	finite state machine
46	When client process has no more data to send issues an	active close	passive close	full close	half close	active close
47	In ____ protocol host uses a window for outbound communication.	UDP	FTP	Sliding window protocol	SMTP	Sliding window protocol
48	RTO means _____	Remote Time out	Retransmission Time in	Retransmission Time in	Remote timing	Retransmission Time in
49	One of the algorithms used in TCP congestion control is _____	Fast Start	Fast Stop	Slow Start	Slow stop	Slow Start

50	_____ defines the size of the buffer in the local TCP	Hardware Type	Protocol size	Software Size	Buffer Size	Buffer Size
51	_____ Protocol returns the Quote of the Day	Quote	Daytime	Users	Discard.	Quote
52	_____ address is used for Multicasting.	Class B	Class D	Class A	Class E	Class D
53	_____ address is used for future purpose	Class B	Class D	Class A	Class E	Class E
54	_____ refers to finding network address	Multihome	Mask	Routing	None	Multihome
55	Vaiable length block is used in _____ address	Class address	classless addressing	Classfl address	None	classless addressing
56	The two terms often used in classless addressing is _____ and _____	address, type	length, prefix	Prefix, prefix length	suffix, suffix length.	Prefix, prefix length
57	In fixed length subnetting, the number of subnets is power of _____	4	5	3	2	5
58	Occasionally used term in classless addressing are _____ and _____	address, type	length, prefix	Prefix, prefix length	suffix, suffix length.	suffix, suffix length.
59	FTP uses the service of _____	IP	TCP	SMTP	IGMP	TCP
60	In FTP Port 21 is used for _____	Control connection	Data connection	Transformation connection	one	Control connection
61	FTP uses_____ character set	NVTA ASCII	VTM ASCII	Binary	None	VTM ASCII
62	_____ mode data is delivered from FTP to TCP as continuous stream of bytes.	Block mode	Compress mode	Stream mode	None.	Compress mode
63	_____ Command terminates the message in SNMP	END	QUIT	LOOP	None	QUIT
64	_____ is a permanent negative completion reply	4YZ	5YZ	3YZ	YZ	3YZ

SYLLABUS

Unit V: Remote Login- FTP – SMTP – SNMP. IP over ATM Wan – Cells – Routing the Cells – ATMARP – Logical IP SUBNETS – VPN.

1. FILE TRANSFER PROTOCOL (FTP)

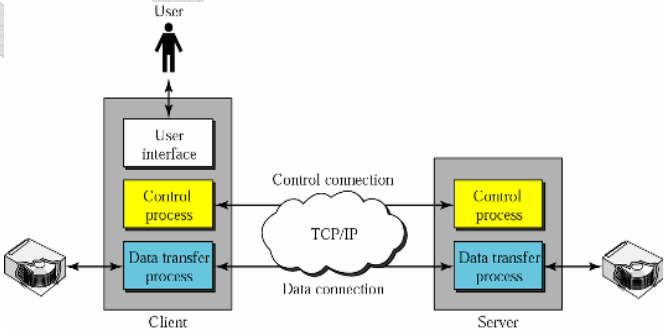
File Transfer Protocol (FTP)

It is the standard mechanism provided by TCP\IP for copying a file from one host to another. Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. All of these problems have been solved by FTP in a very simple and elegant approach. FTP differs from other client-server applications in that it establishes two connections between the hosts. One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient. The control connection uses very simple rules of communication.

We need to transfer only a line of command or a line of response at a time. The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.

FTP uses two well-known TCP Ports: Port 21 is used for the control connection, and port 20 is used for the data connection.

Figure 19.1 shows the basic model of FTP. The client has three components: user interface, client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer process. The control connection is made between the control processes. The data connection is made between the data transfer processes.



The **control connection** remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transferred. It opens each time commands that involve

transferring files are used, and it closes when the file is transferred. In other words, when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

Connections

The two FTP connections control and data use different strategies and different port numbers.

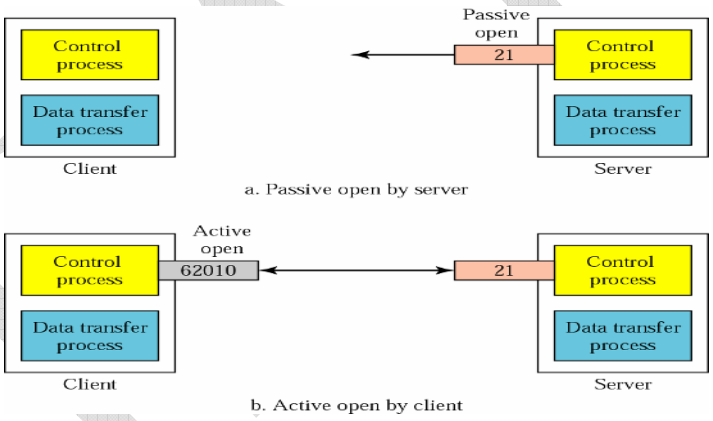
Control Connection

The control connection is created in the same way as other application programs described so far. There are two steps:

1. The server issues a passive open on the well-known port21 and waits for a client.
2. The client uses an ephemeral port and issues an active open.

The connection remains open during the entire process. The service type, used by the IP protocol, is *minimizing delay* because this is an interactive connection between a user (human) and a server. The user types commands and expects to receive responses without significant delay. Figure 19.2 shows the initial connection between the server and the client.

Figure Opening the control connection



Data Connection

The **data connection** uses the well-known port 20 at the server site. However, the creation of a data connection is different from what we have seen so far. The following shows how FTP creates a data connection:

1. The client, not the server, issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.

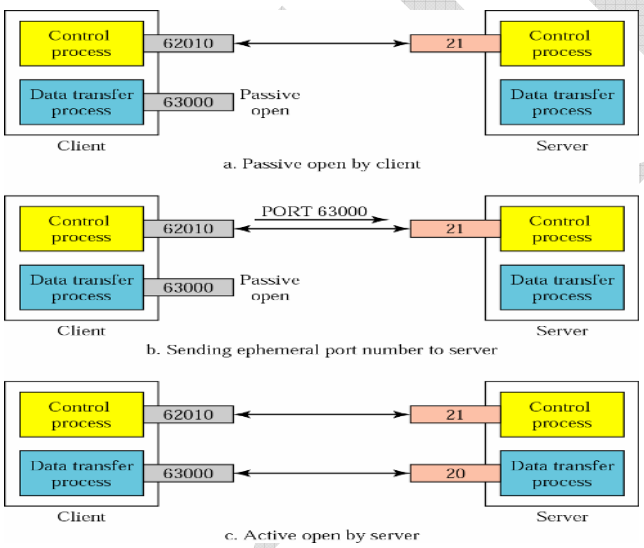
2. The client sends this port number to the server using the PORT command (we will discuss this command shortly).
3. The server receives the port number and issues an active open using the well known port 20 and the received ephemeral port number.

Communication

The FTP client and server, which run on different computers, must communicate with each other. These two computers may use different file formats. FTP must make this heterogeneity compatible.

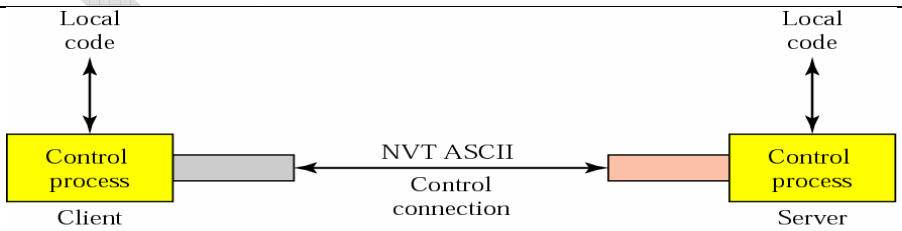
` FTP has two different approaches, one for the control connection and one for the data communication. We will study each approach separately.

Figure creating the data connection



Each line is terminated with a two-character (carriage return and line feed) end-of-line token.

Figure using the control connection

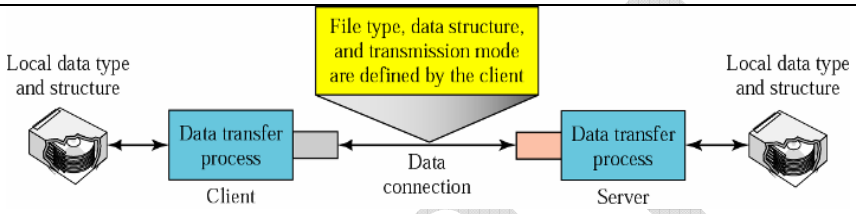


Communication over Data Connection

The purpose and implementation of the data connection are different from that of the control connection. We want to transfer files through the data connection. The client must define the type of file

to be transferred, the structure of the data, and the transmission mode. Before sending the file through the data connection, we prepare for transmission through the control connection. The heterogeneity problem is resolved by defining three attributes of communication: file type, data structure, and transmission mode (see Figure 19.5).

Figure 19.5 Using the data connection



File Type FTP can transfer one of the following file types across the data connection:

- **ASCII file.** This is the default format for transferring text files. Each character is encoded using NVT ASCII. The sender transforms the file from its own representation into NVT ASCII characters and the receiver transforms the NVT ASDCII characters to its own representation.
- **EBCDIC file.** If one or both ends of the connection use EBCDIC encoding, the file can be transferred using EBCDIC encoding.
- **Image file.** This is the default format for transferring binary files. The file is sent as continuous streams of bits without any interpretation or encoding. This is mostly used to transfer binary files such as compiled programs.

If the file is encoded in ASCII or EBCDIC, another attribute must be added to define the printability of the file.

Nonprint. This is the default format for transferring a text file. The file contains no vertical specifications for printing. This means that the file cannot be printed without further processing because there are no characters to be interpreted for vertical movement of the print head. This format is used for files that will be stored and processed later.

- **File structure (default).** The file has no structure. It is a continuous stream of bytes.
- **Record structure.** The file is divided into records. This can be used only with text files.
- **Page structure.** The file is divided into pages, with each page having a page number and a page header. The pages can be stored and accessed randomly or sequentially.

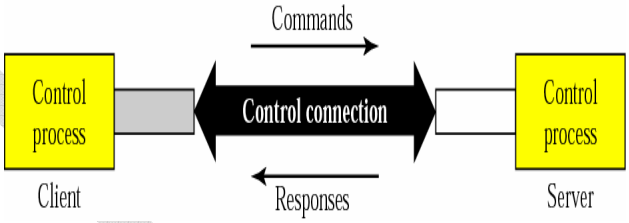
Transmission mode FTP can transfer a file across the data connection using one of the following three transmission modes:

- **Stream mode.** This is the default mode. Data are delivered from FTP to TCP as a continuous stream of bytes. TCP is responsible for chopping data into segments of appropriate size. If the data is simply a stream of bytes (file structure), no end-of-file is needed. End-of-file in this case is the closing of the data connection by the sender. If the data is divided into records (record structure), each record will have a 1-byte end-of-record (EOR) character and the end of the file will have a 1-byte end-of-file (EOF) character.
- **Block mode.** Data can be delivered from FTP to TCP in blocks. In this case, each block is preceded by a 3-byte header. The first byte called the *block descriptor;* the next two bytes define the size of the block in bytes.
- **Compressed mode.** If the file is big, the data can be compressed. The compression method normally used is run-length encoding. In this method, consecutive appearances of a data unit are replaced by one occurrence and the number of repetitions. In a text file, this is usually spaces (blanks). In a binary file, null characters are usually compressed.

Command processing

FTP uses the control connection to establish a communication between the client control process and the server control process. During this communication, the commands are sent from the client to the server and the responses are sent from the server to the client (see figure 19.6).

Figure 19.6 Command processing



Commands

Commands, which are sent from the FTP client control process, are in the form of ASCII uppercase, which may or may not be followed by an argument. We can roughly divide the commands into six groups: access commands, file management commands, data formatting commands, port defining commands, file transferring commands, and miscellaneous commands.

- **Access commands.** These commands let the user access the remote system. Table lists common commands in this group

Command	Arguments(s)	Description
USER	User id	User information
PASS	User password	Password
ACCT	Account to be changed	Account information
REIN	Re install	Reinitialize
QUIT	Terminate	Log out of the system
ABOR	Cancel	Abort the previous command

- **File management commands.** These commands let the user access the file system on the remote computer. They allow the user to navigate through the directory structure, create new directories, delete files, and so on. Table 19.2 gives common commands in this group.

  Table File management commands

Command	Argument(s)	Description
CWD	Directory name	Change to another directory
CDUP		Change to the parent directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories of files
NLIST	Directory name	List the names of subdirectories or files without other attributes
MKD	Directory name	Create a new directory
PWD		Display name of current directory
RMD	Directory name	Delete a directory
RNFR	File name (old file name)	Identify a file to be renamed
RNTO	File name (new file name)	Rename the file
SMNT	File system name	Mount a file system

- **Data formatting commands.** These commands let the user define the data structure, file type, and transmission mode. The defined format is then used by the file transfer commands. Table 19.3 shows common command in this group.

Table Data formatting commands

Command	Argument(s)	Description
Type	A(ASCII),E(EBCDIC),I(Image), N(Nonprint), or T (TELNET)	Define the file type and id necessary the print format
STRU	F(File),R(Record),or P(page)	Define the organization of the data
MODE	S(stream), B(Block), or C(Compressed)	Define the transmission mode

Port defining commands. These commands define the port number for the data connection on the client site. There are two methods to do this. In the first method, using the PORT command, the client can choose an ephemeral port number and send it to the server using passive open. The server uses that port number and creates an active open. In the second method, using PASV command, the client just asks the server to first choose a port number. The server does a passive open on that port and sends the port number in the Response (see response numbered 227 in Table 19.7). The client issues an active open using that port number. Table Port defining commands

Command	Argument(s)	Description
PORT	6-digit identifier	Client chooses a port
PASV		Server chooses a port

File transfer commands. These commands actually let the user transfer files. Table 19.5 lists common commands in this group.

Command	Argument(s)	Description
RETR	File name(s)	Retrieve files; file(s) are transferred from server to the client
STOR	File name(s)	Store files; file(s) are transferred from the client to the server
APPE	File name(s)	Similar to STOR except if the file exists, data must be appended to it

- **Miscellaneous commands.** These commands deliver information at the FTP user at the client site. Table 19.6 shows common commands in this group.

Table 19.6 Miscellaneous commands

Command	Argument(s)	Description
HELP		Ask information about he server
NOOP		Check if server is alive
SITE	Commands	Specify the site-specific commands
SYST		Ask about operating system used by the server

Responses

Every FTP command generates at least one response. A response has two parts: a three digit number followed by text. The numeric part defines the code; the text part defines needed parameters or extra explanations. We represent the three digits as xyz. The meaning of each digit is described below.

First digit The first digit defines the status of the command. One of five digits can be used in this position:

- **1yz (positive preliminary reply).** The action has started. The server will send another reply before accepting another command.
- **2yz (positive completions reply).** The action has been completed. The server will accept another command.

- **3yz (positive intermediate reply).** The command has been accepted, but further information is needed.
- **4yz (transient negative completion reply).** The action did not take place, but he error is temporary. The same command can be sent later.
- **5yz (permanent negative completion reply).** The command was not accepted and should not be retried again.

Second Digit The second digit also defines the status of the command. One of six digits can be used in this position:

- **X0z (syntax).**
- **X1z (information).**
- **X2z (connections).**
- **X3z (authentication and accounting).**
- **X4z (unspecified)**
- **X5z (file system).**

Third digit The third digit provides additional information. Table shows a brief list of possible responses (using all three digits).

Code	Description
Positive Preliminary Reply	
120	Service will be ready
125	Data connection open; data transfer will start shortly
150	File status is OK; data connection will be open shortly
Positive completion reply	
200	Command OK
211	System status or help reply
212	Directory status
213	File status
214	Help message
215	Naming the system type(operating system)
220	Service ready
221	Service closing

225	Data connection open
226	Closing data connection
227	Entering passive mode; server sends its IP address and port number
230	User login OK
250	Request file action OK
Positive intermediate reply	
331	User name OK; password is needed
332	Need account for logging
350	The file action is pending; more information needed
Transient negative completion reply	
425	Cannot open data connection
426	Connection closed; transfer aborted
450	File action not taken; file not available
451	Action aborted; local error
452	Action aborted; insufficient storage
Permanent negative completion reply	
500	Syntax error; unrecognized command

Table Responses (continued)

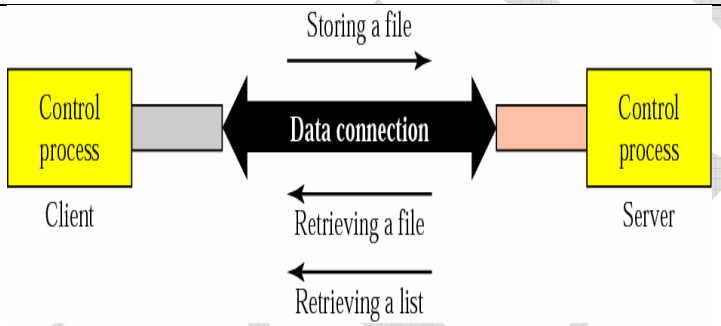
Code	Description
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command parameter not implemented
530	User not logged in
532	Need account for storing file
550	Action is not done; file unavailable
552	Requested action aborted; exceed storage allocation
553	Requested action not taken; file name not allowed

File Transfer

File transfer occurs over the data connection under the control of the commands sent over the control connection. However, we should remember the file transfer in FTP means one of three things (see Figure 19.7).

- A file is to be copied from the server to the client. This is called retrieving a file. It is done under the supervision of the RETR command.
- A file is to be copied from the client to the server. This is called storing a file. It is done under the supervision of the STOR command.
- A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the LIST command. Note that FTP treats a list of directory or file names as a file. It is sent over the data connection.

Figure File transfer



Anonymous FTP

To use FTP, a user needs an account (user name) and a password on the remote server. Some sites have a set of files available for public access. To access these files, a user does not need to have an account or password. Instead, the user can use anonymous as the user name and guest as the password.

User access to the system is very limited. Some sites allow anonymous users only a subset of commands. For example, most sites allow the user to copy some files, but do not allow navigation through the directories.

Flow and Error Control

UDP is a very simple, unreliable transport protocol. There is no flow control, and hence no window mechanism. The receiver may overflow with incoming messages. There is no error control mechanism in UDP except for the checksum. This means thus the sender does not know if message has been lost are duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded. The lack of **flow control and error control** means that the process using UDP should provide for these mechanisms.

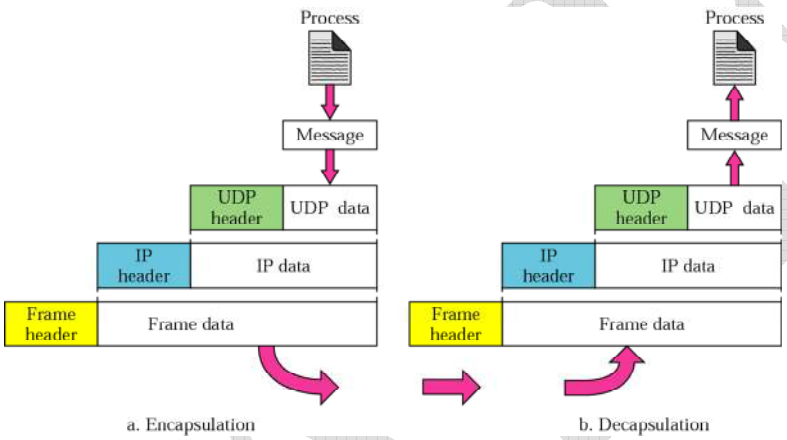
Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages

Encapsulation

When a process has a message to send through UDP, it passes the message to UDP along with a pair of socket addresses and the length of data. UDP receives the data and adds the UDP header.

Figure 11.10 *Encapsulation and decapsulation*



2. Simple Mail Transfer Protocol

Introduction

Internet

Simple Mail Transfer Protocol is a protocol in the vast field of Internet. The Internet is a large network of interconnected computers all around the block. It is a reservoir of information sources and it provides service to everyone who has the access to it throughout the world. The Internet provides a spectrum of services such as electronic mailing, file transfer and file locators. It also act as a platform for many other activities such as publishing documents over the WWW in the form of web pages, conducting business over the net , forming discussion groups and bulletin board services, advertising of services and products. The other services that are provided by the Internet are access to databases. In addition, it allows the user to download enormous information and also provides resource haring among the computers connected on the network.

The hardware part of Internet is just the physical components like the computer either Client or Server, Modem and the connecting media. The hardware is just the body the life to it is given by the software, which is on the Internet. It provides access to the information n the Internet. There are many

software products that are associated with Internet like MOSAIC, INTERNET EXPLORER, NETSCAPE NAVIGATOR etc. these are called web browsers. They are used to help the clients to download, and search the required information on the Internet.

World Wide Web

The Web is an architectural framework for accessing linked documents spread out over thousands of machines all over the Internet. The Web (also known as WWW) began in 1989 at CERN, the European center for nuclear research. CERN has large team of scientists from European countries carrying out research in particles physics. The Web grew out of the need to have these large teams of internationally dispersed researchers collaborate using a constantly changing collection of reports, blueprints, drawings, photos and other documents. Since the Web is basically a Client -Server system this Research Paper discusses both (i.e. .user side and server side [1]).

The Client Side

The Web consists of a vast, worldwide collection of documents, called Web pages. Each page contains links (pointers) to other, related pages, anywhere in the world .Users can follow a link (e.g., by clicking on it), which then takes them to the page pointed to. This process can be repeated indefinitely possibly traversing hundreds of linked pages while doing so. Pages that point to other pages are said to use by hypertext.

Pages are viewed with a program called a browser, of which Mosaic and Netscape are two popular ones. The browser fetches the page requested, interprets the text and formatting commands that it contains and displays the page properly formatted on the screen. Strings of text that are links to other pages, called hyperlinks, are highlighted, either by underlining, displaying them in a special color or both. To follow the link, the user places the cursor on the highlighted are (using the mouse or the arrow keys) and select it (by clicking a mouse button or hitting ENTER).

In addition to having ordinary text (not underlined) and hypertext (underlined), Web pages can also contain icons, line drawings, maps and photographs. Each of these can (optionally) be linked to another page. Clicking on one of these elements causes the browser to fetch the linked page and display it, the same as clicking on text .With images such as photos and maps , which page is fetched next may depend on what part of the image was clicked on.

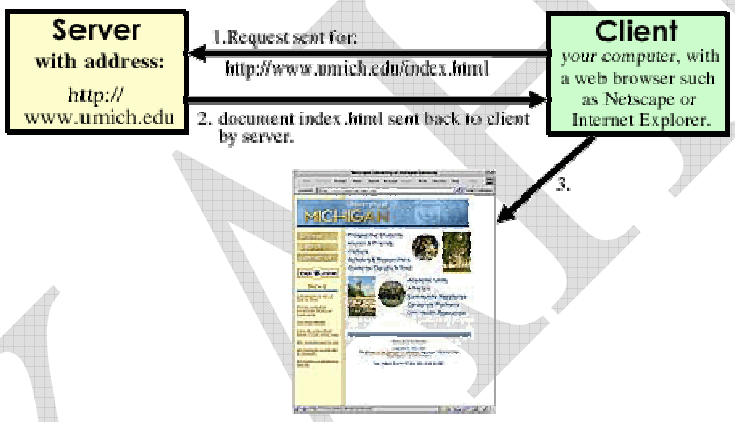
Not all pages are viewable in the conventional way. For example, some pages consist of audio tracks, video clips or both. When hypertext pages are mixed with other media, the result is called hypermedia. To host a web browser, a machine must be directly on the internet or at least have a SLIP (Serial Line IP) or PPP (Point –to-Point Protocol) connection to a router or other machine that is directly

on the Internet. This requirement exists because the way a browser fetches a page is to establish a TCP connection to the machine where the page is, and then send a message over the connection asking for the page. If it cannot establish a TCP connection to an arbitrary machine on the Internet, a browser will not work

The Server Side

Every website has a server process listening to TCP port 80 for incoming connections from clients (normally browsers). After a connection has been established, the client sends one request and the server sends one reply. Then the connection is released. The protocol that defines the legal requests and replies is called HTTP.

The user clicked on some pieces of text or perhaps on the icon those points to the URL (Uniform Resource Locator). URL has three parts: the name of the protocol, the name of machine where the page is loaded and the name of the file containing the page



This is what's displayed through your web browser.

The steps that occur between the user's click and the page being displayed are as follows.

1. The browser determines the URL.
2. The browser asks DNS for the IP address of the website.
3. DNS replies with IP address.
4. The browser makes a TCP connection to port 80 on IP address.
5. If then sends a GET command for getting information.
6. The server sends the requested file.
7. The TCP connection is released.
8. The browser displays all the text in the requested web site.
9. The browser fetches and displays all the images in the requested web site.

Many browsers display which step they are currently executing in a status line at the bottom of the screen. In this way, when the performance is poor, the user can see if it is due to DNS (Domain Name System) not responding, the server not responding, or simply network congestion during page transmission.

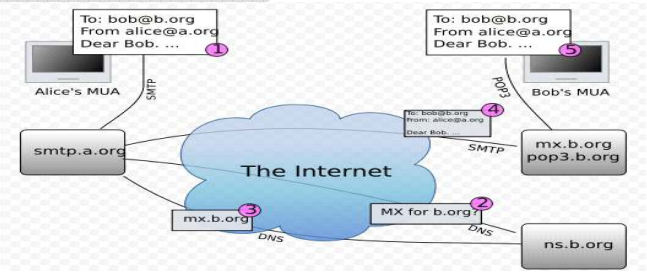
The E-Mail System

In present day life, communication plays a vital role. Some of the traditional communication facilities like postal service, fax, couriers have many drawbacks. These drawbacks are overcome by using the fast growing electronic communication called electronic mail or E-Mail. The term E-Mail simply refers to transfer of text messages from one computer to another over a type of network. E-mail can be used to send spread sheet files, word processing, documents, programs, images, voice messages and faxes. Almost anything can be stored electronically and can be sent.

The Working of E-Mail

To send a message from local computer to another user through E-Mail system, the client software creates the message in a temporary storage space. If the user tells the system to send the message, the system transfers that to a message database on the mail server. This Message is stored in a database in encrypted format. Thus casual browsing does not reveal its content. The mail system increments a counter in a recipient's mailbox to indicate the user of the new mail's presence.

The next time that the user checks his mail, he would see that he has one new message. If he issues the command to read that message, the mail system decrypts the message and sends it to his PC for display. If he chooses to save the message, it remains in the message database in its encrypted form. On the other hand, if he decides to delete it, the mail system remove it from the message database and decrements the counter in his mailbox. If he decides to save a cop to his local hard disk, it saves it in decrypted form.



Addressing and Standards

Addressing an e-mail message means putting information in the header that will enable the sending and receiving computers to deliver the message correctly. There are lots of different e-mail

systems, each with its own addressing skills. The addressing skills depend on network protocol. [3] E - mail would look pretty straightforward. There are different e-mail systems, each with its own addressing schemes.

The Internet Standard

Most of the WAN e-mail sent within North America is arranged in the form at known as Internet format. Because the Internet is the largest and most well known network, which uses this format, hence it's often called the internet format, but the actual name of the standard is RFC – 822. Examples of e-mail addresses that use this format are

Mrobbins@bga.com

Etittel@zilker.net

The naming system on the internet is called the Domain Name System abbreviated as DNS. Thus, addresses in the Internet format are also called DNS addresses. DNS addressing will be the only one standard on which the world will settle, largely because there is already a considerable volume of software that uses it; and also it has been deployed in so many networks throughout the words.

Store and Forward

There are two types of messaging systems. They are

1. Real –time systems
2. Store –and-forward systems.

In the real-time systems one can send messages only to users who are currently logged in to the system. In Store-and-Forward systems messages are held up in a central repository until users retrieve them. This system also queues up mail coming from different networks and sends groups of messages in batches, instead of sending each message as soon as it arrives. The delays introduced by intermediate storage are usually invisible to the users.

A big advantage of store-and-forward systems is that they are less vulnerable to network problems than real time systems. If a link foes down in a real-time-system, no message can get through until the link recovers. If a link goes down in a store-and-forward system, the system queue incoming messages and delivers after the link are back up.

Mailbox

A mailbox is a place where the e-mail system stores messages for that user. It contains messages received and read messages, which are not yet deleted. The user needs to clean out his mailbox from time to time by deleting unwanted messages because storing in his mailbox would clog the mailbox.

E-Mail Architecture

E-mail systems consist of two subsystems: the user agents, allows people to read and send e-mail, and the message transfer agents, moves the messages from the source to the destination. The user agents are local programs that provide a command-based, menu based or graphical method for interacting with the email system. The message transfer agents are typically system daemons that run in the background and move email through the system.

E – Mail Using Browser

Most users log on to a web page using corresponding web servers for sending mails. As the client requests for that page the web server responds using HTTP protocol. For sending mails, the browser used by the client gives the date to the web server, which in turn gives to the middle ware. The middle ware converts the data into a mail format which mail server can understand. To make the mail server understand the data, it should be converted into commands [3].

SMTP commands are used for sending mails. So the data from the web server are converted into various SMTP commands by the middle ware (Java Server Pages [JSP], Active Server Pages [ASP]) used. These commands are given to the mail server and the responses are evaluated. This mail server is called as Sender SMTP server. This server transfers the mails to the corresponding destination servers (Fig.1.2).

E-Mail Using Mail Client

Mail client software is designed in such a way that it abides the protocols. As mail client follows the protocols, for sending mails the SMTP commands are directly send to the source mail server. This in turn passes the mail to the destination mail server. (Fig.1.3)[3]

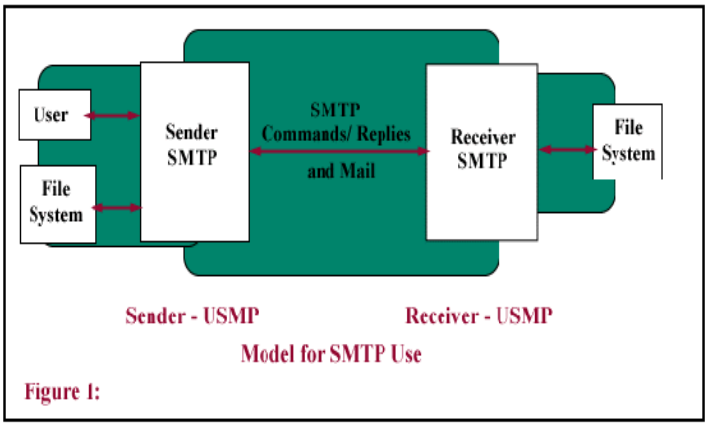


Fig.1.3 E-mail Using Mail Client

E – Mail Sending Process

The SMTP provides mechanisms for the transmission of mail directly from the sending user's host to the receiving user's host when the two hosts are connected via one or more SMTP servers. These

transactions triggered by various SMTP commands. The conversation between client and mail server are given below:

Sender: MAILFROM:Smith@Alpha.ARPA

Receiver: 250 OK

Sender: RCPT TO:Jones@Beta.ARPA

Receiver: 260 OK

Sender: DATA

Receiver: 354 Start mail input; end with<CRLF>.<CRLF>

Sender: "God is great" [Message]

Receiver: 250 OK

In this conversation there is no authentication happening. The sender can enter anything at the HELO prompt during identification. The same applies to the MAIL FROM prompt

Services of E-Mail

E-Mail system supports basic functions.

1 Composition refers to the process of creating messages and answers.
2 Transfer refers to moving messages from the originator to the recipient.
3 Reporting has to do with telling the originator what happened to the message.
4 Displaying incoming messages is needed so people can read their email.
5 Disposition is the final step and concerns what the recipient does with the message after receiving it. [1]

Mail Server and Mail Clients

Mail Server Details

Mail server is a software program that distributes files or information in response to requests sent via e-mail. The mail server sends and receives email messages are directed by the client. For those technical types, the Post Office Protocol version 3(POP3) servers receive mail and send them to the client. The Simple Mail Transfer Protocol (SMTP) server sends off e-mails to other computers. The mail server handles mailboxes for any number of users (or purposes).

These mailboxes can be accessed by the POP3 protocol , which is currently the most popular method for mail access ,using this mail's can be left on the server or downloaded to the user's own machine.

A server is used to store mail in a store-and-forward architecture. Workstations send mail to a mail server for forwarding, and retrieve any mail that might have been received from their electronic

postbox. Powerful administration facilities allow the automatic expiry of mail that has been left in the mailbox for too long, it also reduces the resource Demand without having to persuade the users to do these themselves.

Courier Mail Server

The courier mail transfer agent (MTA) is an integrated Mail/Group Ware server based on open commodity protocols, such as IM AP (Internet Mail Accessing protocol), POP3, SSL (Secure Socket Layer), LDAP (Lightweight Directory Access Protocol) and HTTP. Courier provides IMAP, POP3, Web Mail, and mailing lists services within a single, consistent, framework. Courier's source code should compile on POSIX- based operating systems based on Linux, and BSD – Derived Kernels. Courier should also compile on Solaris and AIX, with some help from Sun or IBM's freeware add-on tools for their respective operating systems. [17]. Courier implements SMTP extensions for mailing list management and spam filtering , Courier can function as an intermediate mail relay , relaying mail between an internal LAN and the Internet , or perform final delivery to mailboxes. Courier's configuration is set by plain text files and Perl scripts.

Courier can also provide mail services for operating system accounts, virtual mail accounts, managed by an LDAP, My SQL, or Postures-based authentication database. Certain portions of Courier - the mail filtering engine, the Web mail server and the IMAP server are also available in separate, smaller, packages that can be used with other mail servers.

Commands in SMTP protocol

Command Semantics

The SMTP commands define the mail transfer or the mail system function requested by the user. SMTP commands are character strings terminated by <CRLF>. The command codes themselves are alphabetic characters terminated by <SP> if parameters follow and <CRLG> otherwise. The syntax of mailboxes must conform to receiver site conventions. The SMTP commands are discussed below

A mail transaction involves several data objects, which are communicated as arguments to different commands. The reverse – path is the argument of the MAIL command, the forward –path is the argument of the RCPT command, and the mail data is the argument of the DATA command. Those arguments or data objects must be transmitted and held pending till the confirmation is communicated by the end of email, which finalizes the transaction. The model for this is that distinct buffers are provided to hold the types of data objects, that is, there is a reverse – path buffer, a forward-path buffer, and a mil data buffer. Specific commands cause information to be appended to a specific buffer, or cause one or more buffers to be cleared.

Hello (HELO)

This command is used to identify the sender – SMTP to the receiver –SMTP. The arguments field contains the host name of the sender-SMTP. The receiver – SMTP identifies itself to the sender-SMTP through the greeting reply, and through then response to this command. This command and an OK reply to it confirm that both the sender –SMTP and the receiver-SMTP are in the initial state, that is, there is no transaction in progress and all state tables and buffers are cleared.

Mail (MAIL)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more mailboxes. The argument field contains a reverse-path. The reverse-path consists of an optional list of hosts and the sender mailbox. When the lists of hosts are present, it is a "reverse" source route and indicates that the mail was relayed through each host on the list (the first host in the last was the most recent relay). This list is used as a source route to return non-delivery notices to the sender. As each relay host adds itself to the beginning of the list , it must use its name as known in the IPCE to which it is relaying the mail rather than the IPCE from which the mail came.(if they are different). In some types or error reporting messages (for e.g.., undeliverable mail notification) i.e... The reverse –path may be null.

This command cleans the reverse – path buffer, the forward – path buffer, and the mail data buffer; and inserts the reverse – path information from this command into the reverse-path buffer. This command is used to identify an individual recipient of the mail data; multiple recipients are specified by multiple use of this command. The forward – path consists of an optional list of hosts and a required destination mailbox. When the list of hosts is present, it is a source route and indicates that the mail must be relayed to the next host on the list. If the receiver – SMTP does not implement the relay function it may be send the user the same reply.

When the mail is relayed, the relay host must remove from the beginning forward – path and put itself at the beginning of the reverse- path. When the mail reaches its ultimate destination (the forward path contains only the destination mail box,) the receiver SMTP inserts it into the destination mail box in accordance with its host mail conventions.

For egg; mail received at the relay host A with arguments

FROM : <USERX@HOSTY.ARPA

TO :< @HOSTA.ARPA, @HOSTB.ARPA:USERC@HOSTD.ARPA>

Will be relayed on to host B with arguments

FROM :< @HOSTA.ARPA:USERX@HOSTY.ARPA>

TO: <@HOSTB.ARPA:USERC@HOSTD.ARPA>

Data (DATA)

The receiver treats the following the command as mail data from the sender. This command causes the mail data from this command to be appended to the mail data buffer

The mail data may contain any of the 128 ASCII character codes. The mail data is terminated by a line containing only a period. , that the character sequence "<CRLF>, <CRLF>. This is the end of the mail data indication.

The end of the mail data indication that the receiver must now process the stored mail transaction information. This processing consumes the information in the reverse-path buffer, the forward path buffer and the mail data buffer, and the completion of this command this buffers are cleared. If the processing, is successful the receiver must send an OK reply. If the processing is fails completely receiver must send a failure reply.

When the receiver – SMTP accepts a message either for relaying or for final delivery it inserts at the beginning of the mail data a time stamp line. The time stamp line indicates the identity of the host machine that send the message, and identity that receive the message (and is inserting the time stamp), and the time and date the message was received. Relayed message will have multiple time stamp lines.

When the receiver-SMTP makes the 'final delivery' of a message it inserts at the beginning of the mail data a return path line. The return path line preserves the information in the <reverse-path> from the mail command. Here, final delivery means the message leaves the SMTP world. Normally, this would be meaning it has been delivered to the destination user, but in some cases it may be further processed and transmitted by another mail system. It is possible for the mailbox in the return path be different from the actual sender's mailbox, for example if error responses are to be delivered a special error handling mailbox rather than the message senders.

The preceding two paragraphs imply that the final mail data will begin a return path line, followed by one or more time stamp lines. Special mention is needed of the response and further action required when the processing following the end of the mail data indication is partially successful. This could arise if after accepting several recipients and the mail data , the receiver-SMTP finds that the mail data can be successfully delivered to some of the recipients , but it cannot be to others (for egg ; due to mailbox space allocation problems). In such a situation, the response to the DATA command must be an OK reply. But, the receiver –SMTP must compose and send an "undeliverable mail" notification message to the originator of the message. Either a single which lists all of the notifications which lists all of the recipients that failed to get the message, or separate notification messages are sent using the MAIL command.

Example of RETURN path and RECEIVED time stamps

Return path :<@GHI.ARPA,@DEF.ARPA,@ABC.ARPA:JOE@ABC.ARPA>
Received : from GHI.ARPA by JKL.ARPA ; 27 OCT 81 15:27:39 PST
Received : from DEF.ARPA BY GHI.ARPA ; 27 OCT 81 15:15:13 PST
Received :from ABC.ARPA by DEF.ARPA ; 27 OCT 81 15:01:59 PST
Date : 27 OCT 81 15:01:01 PST
From :JOE@ABC.ARPA
Subject : Improved mail system installed
To :SAM@JKL.ARPA

Send(SEND)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more terminals. The argument field contains a reverse path. This command is successful if the message is delivered to a terminal. The reverse path consists of an optional list of hosts and the sender mailbox. When the list of hosts is present, it is a reverse source route and indicates that the mail was relayed through each host on the list (the first host in the list was the most recent relay).

This list is used as a source route to return non-delivery notices to the sender , as each relay host adds itself to the beginning of the list , it must use its name as known on the IPCE to which it is relaying the mail than the IPCE from which the mail came (if they are different).

This command clears the reverse path buffer, the forward path buffer, and the mail data buffer, and inserts the reverse path information from this command into the reverse path buffer.

Verify (VRFY)

This command asks the receiver to conform that the argument identifies a user. If it is a user name, the full name of the user (if known) and the fully specified mailbox are returned. This command has no effect on any of the reverse path buffer, the forward path buffer, or the mail data buffer.

Quit (QUIT)

This command specified that the receiver must send an OK reply and then close the transmission channel. The receiver should not close the transmission channel until it receives and replies to a QUIT command. The sender should not close the transmission channel until it sends a QUIT command and receives the reply. If the connection is closed prematurely the receiver should acts as a RSET command had been received (canceling any pending transaction, but not undoing any previously completed transaction), the sender should act as if the command or transaction in progress had received a temporary error.

Command Syntax

The command consists of a command code followed by an argument field. Command codes are four alphabetic characters. Upper and lower case alphabetic characters are to be treated identically. Thus, any of the following may represent the mail command: MAIL mail Mall mail

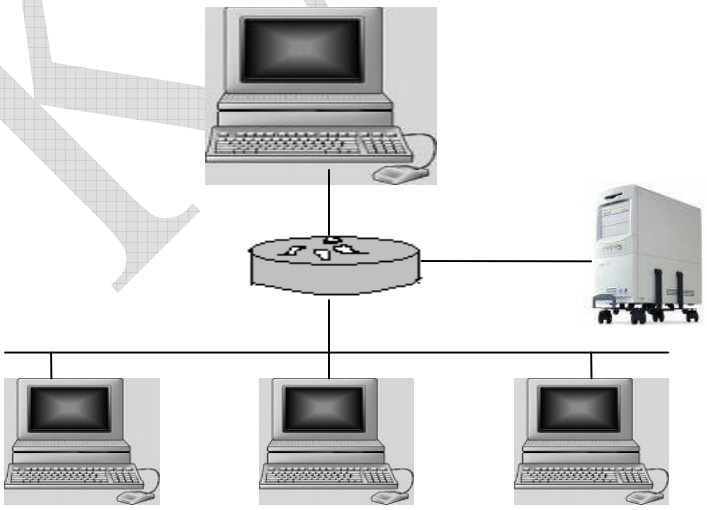
This also applies to any symbols representing parameter values such as "TO" or "to" for the forward path. Command codes and the argument fields are separated by one or more spaces. However, with in the reverse path and forward path arguments case is important. In particular, in some hosts the user "smith" is different from the user "SMITH".

***3.* Simple Network Management Protocol**

Background

The Simple Network Management protocol (SNMP) is an application layer protocol that facilitates the exchange of the management information between network devices. It is part of the Transmission Control Protocol / Internet Protocol (TCP/IP) protocol suite. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth.

Two versions of SNMP exist: SNMP version 1 (SNMPv1) and SNMP version 2 (SNMPv2). Both versions have a number of features in common. but SNMPv2 offers enhancements , such as additional protocol operations. Standardization of yet another version of SNMP - SNMP version 3 (SNMPv3) – is pending. This chapter provides descriptions of the SNMPv1 and SNMPv2 protocol operations. Figure 56-1 illustrates a basic network managed by SNMP.

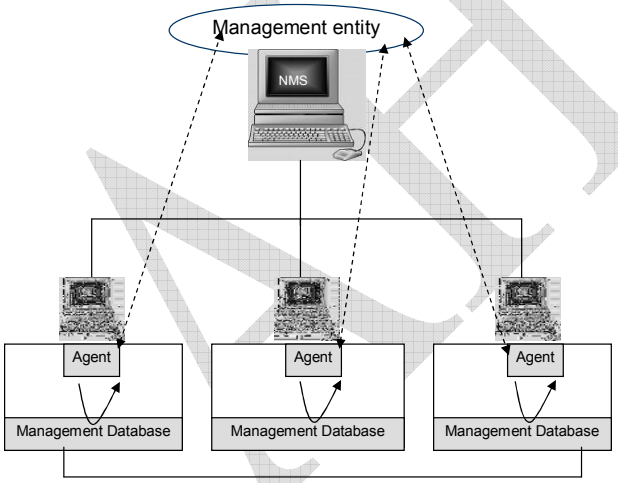


SNMP Basic Components

An SNMP - managed network consists of three key components: managed devices, agents, and network – management systems (NMSs).

A managed device is a network node that contains an SNMP agent and that resides on a managed network. Managed devices collect and store management information and make this information available to NMSs using SNMP. Managed devices, sometimes called network elements, can be routers and access servers, switches and bridges, hubs, computer hosts, or printers.

An agent is a network management software module that resides in a managed device. An agent has local knowledge of management information and translates that information into a form compatible with SNMP. An NMS executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. One or more NMSs must exist on any managed network.

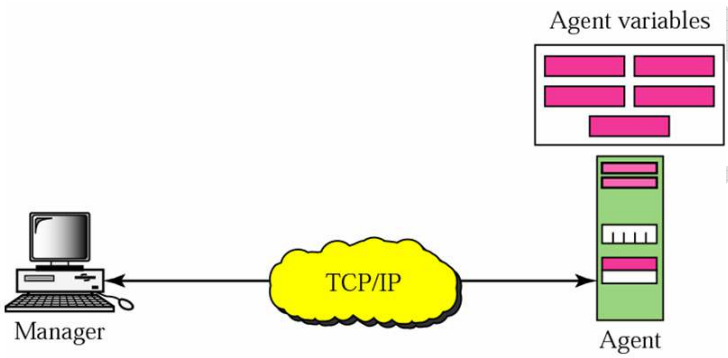


SNMP Commands

Managed devices are monitored and controlled using four basic SNMP commands: read, write, trap, and traversal operations. The read command is used by an NMS to monitor managed devices. The NMS examines the different variables that are maintained by the managed devices. The write command is used by an NMS to control managed devices. The NMS changes the values of the variables stored within managed devices. The trap command is used by the managed devices to asynchronously report events to the NMS. When certain types of events occur, a managed device sends a trap to the NMS. Traversal operations are used by the NMS to determine which variables a managed device supports and to sequentially gather information in variable tables, such as a routing table.

Network Management Architecture

Network management system contains two primary elements. A manager and agents. The manager is the console through which the network administrator performs network management functions. Agents are the entities that interface to the actual device being managed. Bridges, hubs, routers or network servers are examples of managed devices that contain managed objects. These managed objects might be hardware, configuration parameters, performance statistics, and so on, that directly relate to the current operation of the device in question. These objects are arranged in what is known as a virtual information database, called a management information base, also called MIB. SNMP allows managers and agents to communicate for the purpose of accessing these objects. The model of network management architecture looks like this:



Typical agent usually:

1 Implements full SNMP protocol.

2 Stores and retrieves management data as defined by the MIB.

3 Can asynchronously signal an event to the manager.

4 Can be a proxy for some non-SNMP manageable network node. Click here to see typical proxy architecture.

Atypical manager usually:

1 Implemented as a Network Management Station (the NMS)

2 Implements full SNMP protocol

3 Able to send Query

4 Get responses from agents

5 Set variables in agents

6 Acknowledge asynchronous events from agents

Some prominent vendors offer network management platforms which implement the role of the manager (listed in alphabetic order):

1. Dec PolyCenter Network Manager
2. Hewlett – Packard Open View
3. IBM AIX NetView/6000
4. SunConnect SunNet Manager

Management Information Base

Management Information Bases (MIBs) are a collection of definitions, which define the properties of the managed object within the device to be managed. Every managed device keeps a database of values for each of the definitions written in the MIB. It is not the actual database itself – it is the implementation dependent. Definition of the MIB conforms to the SMI given in RFC 1155. Latest Internet MIB is given in RFC 1213 sometimes called the MIB-II. Click here to see MIB architecture. You can think of a MIB as an information warehouse.

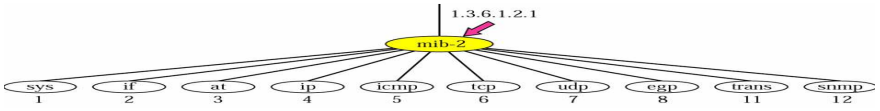
Criteria and Philosophy for standardized MIB

1 Objects have to be uniquely named
2 Objects have to be essential
3 Abstract structure of the MIB needed to be universal
4 For the standard MIB maintain only a small number of objects
5 Allow for private extensions
6 Object must be general and not too device dependent
7 Objects cannot be easily derivable from their objects
8 If agent is to be SNMP manageable then it is mandatory to implement the Internet MIB (currently given as MIB-II in RFC 1157)

Naming an object

1. Universal unambiguous identification of arbitrary objects
2. Can be achieved by using an hierarchical tree
3. Based on the Object Identification Scheme defined by OSI

The Registered Tree



Identifiers

1 Object name is given by its name in the tree.

2 All child nodes are given by the unique integer values within the new sub-tree.

3 Children can be parents of further child sub-tree (ie: they have subordinates) where the numbering scheme is recursively applied.

4 The Object Identifier (or name) of an object is the sequence of non-negative Integer values traversing the tree to the node required.

5 Allocation of an integer value for a node in the tree is an act of registration by whoever has delegated authority for that sub tree.

6 This process can go to an arbitrary depth.

7 If a node ha children then it is an aggregate node.

8 Children of the same parent cannot have the same integer value.

Object and Object Identifiers

1 Object is named or identified by the sequence of integers in traversing the tree to the object type required

2 This does not identify an instance of the object

3 The Object Identifier(OID) is shown in afew ways with a.b.c.d.e being the preferred

4 OIDs can name many types of objects:

The Internet Sub – tree

- Directory sub-tree if for future directory services
- Experimental sub-tree is for experimental MIB work – still
- Has to be registered with the authority (IESG)
- MIB sub-tree is the actual mandatory Internet MIB for all
- Agents to implement (currently MIB-II RFC 1156- this is the Only sub-tree for management)
- Enterprise sub-tree (of private) are MIBs of proprietary objects And are of course not mandatory (sub-tree registered with Internet assigned numbers authority) for example: CISCO
- Router OID: 1.3.6.1.4.1.9.1.1
- SNMP management nearly always Internet in MIB and specific enterprises MIBs.

MIB-II Standard Internet MIB

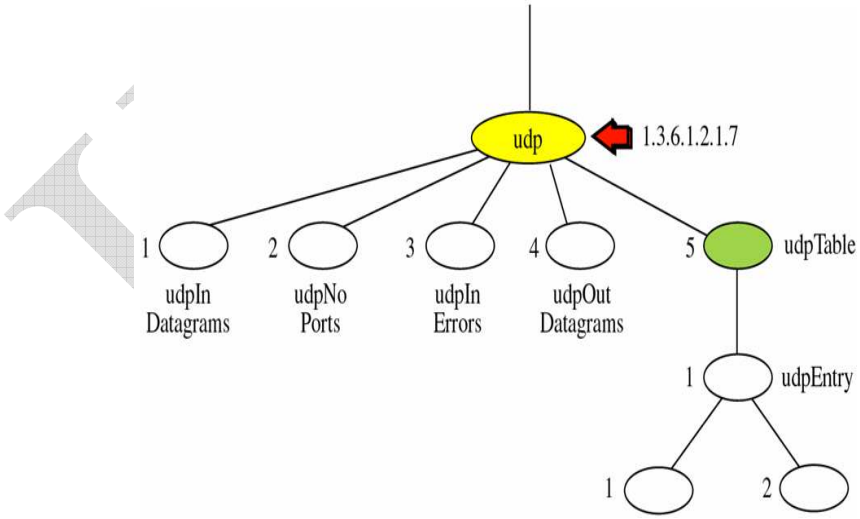
1. Definition follows structure given in SMI
2. MIB-II (RFC 1213) is current standard definition of the virtual file store for SNMP manageable objects

3. Has 10 basic groups
    - System
    - Interfaces
    - At
    - Ip
    - Icmp
    - Tcp
    - Udp
    - Egp
    - Transmission
    - Snmp

If agent implements any group then is has to implement all of the managed objects within the group. An agent does not have to implement all groups. Note: MIB –I and MIB-II have some OID (position in the Internet sub-tree)

MIB-II

The MIB sub-tree



Note: there is an object cm OT (9) under the MIB but it has become almost superfluous and for all intense and purposes is not one of the SNMP manageable groups within MIB.

SNMP Protocol

SNMP is based on the managers/ agent model. SNMP is referred to as "simple" because the agent requires minimal software. Most of the processing power and the data storage reside on the management system, while a complementary subset of those functions resides in the managed system.

To achieve its goal of being simple, SNMP includes a limited set of management commands and responses. The management system issues Get, GetNext and Set messages to retrieve single or multiple object variables or to establish the value of a single variable. The managed agent sends a response message to complete the Get, GetNext or Set. The managed agents send an event notification, called a trap to the management system to identify the occurrence of conditions such as threshold that exceeds a predetermined value. In short there are only five primitive operations:

1 Get(retrieve operation)
2 Getnext(traversal operation)
3 Getresponse(indicative operation)
4 Set(alter operation)
5 Trap(asynchronous trap operation)

SNMP Message Construct

Each SNMP message has the format:

1 Version number
2 Community name – kind of a password
3 One or more SNMP PDUs – assuming trivial authentication

Each SNMP PDU except trap has the following format:

1 Request id – request sequence number
2 Error status – zero if no error otherwise one of a small set
3 Error index – if non zero indicates which of the OIDs in the PDU caused the error 2
4 List of OIDs and values - values are null for get and getnext

Trap PDUs have the following format:

1 Enterprise – identifies the type of object causing the trap
2 Agent address – IP address of agent which sent a the trap
3 Generic trap id – the common standard traps
4 Specific trap id – proprietary or enterprise trap
5 Time stamp – when trap occurred in time ticks
6 List of OIDs and values – OIDs that may be relevant to

Send to the NMS

Outline of the SNMP protocol

1 Each SNMP managed object belongs to a community
2 NMS station may belong to multiple communities
3 A community is defined by a community name which is an OctetString with 0 to 255 octets in length.

Security levels with basic SNMP

Authentication

1 Trivial authentication based on plain text community name exchanged in SNMP message
2 Authentication is based on the assumption that the message is not tampered with or interrogated

Authorization

1 Once community name is validated then agent or manager checks to see if sending address is permitted or has the rights for the requested operation
2 "View" or "cut" of the objects together with permitted access rights is then derived for the pair(community name , sending address)

Summary

1 not very secure
2 SNMP version2 is addressing this
3 Extended security is possible with current protocol (eg: DES and MD5)
4 Does not reduce its power for monitoring

VPN

A virtual private network (VPN) is a technology that creates a safe and encrypted connection over a less secure network, such as the internet. VPN technology was developed as a way to allow remote users and branch offices to securely access corporate applications and other resources. To ensure safety, data travels through secure tunnels, and VPN users must use authentication methods -- including passwords, tokens or other unique identification procedures -- to gain access to the VPN server. VPNs are used by remote workers who need access to corporate resources, consumers who may want to download files and business travelers who may want to log into sites that are geographically restricted. VPN services are critical conduits through which data can be transported safely and securely.

A virtual private network (VPN) is a technology that creates a safe and encrypted connection over a less secure network, such as the internet. VPN technology was developed as a way to allow remote users and

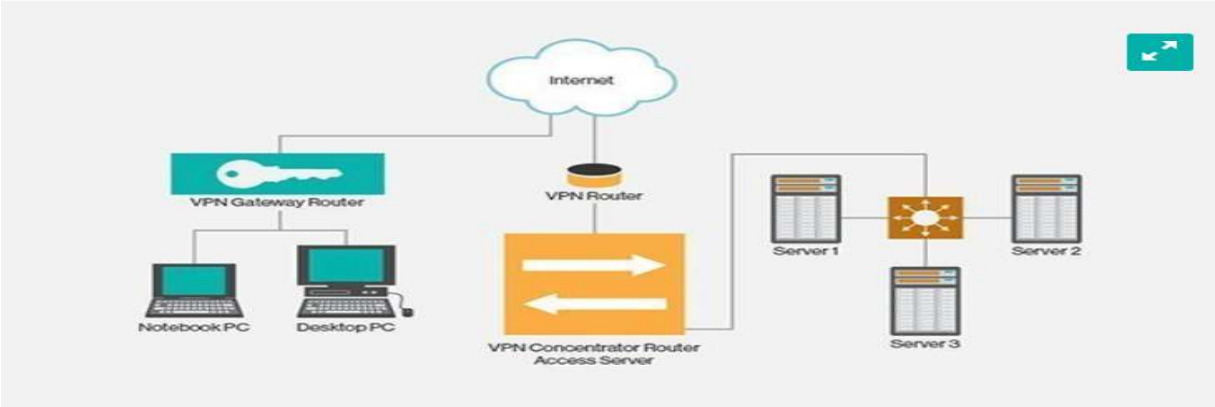
branch offices to securely access corporate applications and other resources. To ensure safety, data travels through secure tunnels, and VPN users must use authentication methods -- including passwords, tokens or other unique identification procedures -- to gain access to the VPN server. VPNs are used by remote workers who need access to corporate resources, consumers who may want to download files and business travelers who may want to log into sites that are geographically restricted. VPN services are critical conduits through which data can be transported safely and securely.

How a VPN works and why you should use one?

The two most common types of VPNs are remote access VPNs and site-to-site VPNs. A remote access VPN uses a public telecommunication infrastructure like the internet to provide remote users with secure access to their organization's network. This is especially important when employees are using a public Wi-Fi hotspot or other avenues to access the internet and connect to their corporate network. A VPN client on a remote user's computer or mobile device connects to a VPN gateway on the organization's network. The gateway typically requires the device to authenticate its identity. Then, it creates a network link back to the device that allows it to reach internal network resources -- e.g., file servers, printers and intranets -- as though the gateway is on the network locally.

A remote-access VPN usually relies on either IP Security (IPsec) or Secure Sockets Layer (SSL) to secure the connection, although SSL VPNs are often focused on supplying secure access to a single application rather than to the entire internal network. Some VPNs provide Layer 2 access to the target network; these require a tunneling protocol like the Point-to-Point Tunneling Protocol or the Layer 2 Tunneling Protocol running across the base IPsec connection. In addition to IPsec and SSL, other protocols used to secure VPN connectivity and encrypt data are Transport Layer Security and OpenVPN.

A site-to-site VPN uses a gateway device to connect an entire network in one location to a network in another -- usually a small branch connecting to a data center. End-node devices in the remote location do not need VPN clients because the gateway handles the connection.



Most site-to-site VPNs connecting over the internet use IPsec. It is also common for them to use carrier MPLS clouds rather than the public internet as the transport for site-to-site VPNs. Here, too, it is possible to have either Layer 3 connectivity (MPLS IP VPN) or Layer 2 (virtual private LAN service) running across the base transport. VPN services can also be defined as connections between specific computers, typically servers in separate data centers, when security requirements for their exchanges exceed what the enterprise network can deliver. Increasingly, enterprises also use VPN connections in either remote access mode or site-to-site mode to connect -- or connect to -- resources in a public infrastructure-as-a-service environment. Newer hybrid-access scenarios put the VPN gateway itself in the cloud, with a secure link from the cloud service provider into the internal network.

Benefits of using a VPN

The justification for using VPN access instead of a private network usually boils down to cost and feasibility: It is either not feasible to have a private network -- e.g., for a traveling sales rep -- or it is too costly to do so.

Possible Questions

Two Mark Questions

1. What is FTP?
2. What is the use of VPN?
3. Who is a User-Agent?
4. What is a Mail-Agent?
5. Define a logical Address.

Six Mark Questions

1. Discuss about the user Agent of mail transfer protocol in detail.
2. What is ARP? Explain about ATMARP with its operations and Format.
3. Discuss about four different scenarios of Mail transfer.
4. How are components managed through SNMP? Discuss in detail.
5. How do IP address switches over ATM? Discuss its basic concepts.
6. Explain the basic concept of FTP.
7. Write a general note on SNMP.
8. Discuss about the negotiation Options of TELNET in detail.
9. What is a TELNET? Discuss about the time-sharing, NVT and Embedding concept of TELNET.
10. What is the use of VPN? Discuss it in general.

Unit V

S.No	Question	Choice1	Choice2	Choice3	Choice4	Answer
1	_____ Protocol returns the Quote of the Day	Quote	Daytime	Users	Discard.	Quote
2	_____ address is used for Multicasting.	Class B	Class D	Class A	Class E	Class D
3	_____ address is used for future purpose	Class B	Class D	Class A	Class E	Class E
4	_____ refers to finding network address	Multihome	Mask	Routing	None	Multihome
5	Vaiable length block is used in _____ address	Class address	classless addressing	Classfl address	None	classless addressing
6	The two terms often used in classless addressing is _____ and _____	address, type	length, prefix	Prefix, prefix length	suffix, suffix length.	Prefix, prefix length
7	In fixed length subnetting, the number of subnets is power of _____	4	5	3	2	5
8	Occasionally used term in classless addressing are _____ and _____	address, type	length, prefix	Prefix, prefix length	suffix, suffix length.	suffix, suffix length.
9	FTP uses the service of _____	IP	TCP	SMTP	IGMP	TCP
10	In FTP Port 21 is used for _____	Control connection	Data connection	Transformation connection	one	Control connection
11	FTP uses_____ character set	NVTA ASCII	VTM ASCII	Binary	None	VTM ASCII
12	_____ mode data is delivered from FTP to TCP as continuous stream of bytes.	Block mode	Compress mode	Stream mode	None.	Compress mode
13	_____ Command terminates the message in SNMP	END	QUIT	LOOP	None	QUIT
14	_____ is a permanent negative completion reply	4YZ	5YZ	3YZ	YZ	3YZ
15	ATM has _____ formats for header	2	1	4	3	1
16	The switches inside he ATM Network route he cells on the _____	VPI	VCI	ARP	None	VPI
17	OPER is a _____bit field	8	45	12	34	45
18	SPA stands for _____	Sender Protocol Address	Seder Protocol Access	Sender Private Application	None	Sender Protocol Address
19	ATM accepts _____ bytes and transfers it into _____ bytes.	40 to 50	30 to 50	48 t65	48 to 53.	48 t65
20	First Phase in the Mobile Communication is _____	Agent advertisement	Agent finding	Agent discovery	None	Agent advertisement
21	_____ Network is designed to be used only inside an organization	Protected	Private	Sensitive	Non Sensitive	Sensitive
22	_____ Networks have its private network with global internet access.	Globalization	Interknitting	Hybrid	None	Interknitting
23	_____ is a simple and efficient adaptation layer.	AAL6	AAL6	AAL3	AAL4	AAL6
24	_____ Command in SMTP is used to ask Receipt to send information about the command as the argument.	TURN	HELP	SEEK	QUIT.	TURN
25	The first and second stage of Mail delivery use _____	FTP	SMTP	SNMP	None	FTP
26	POP stands for _____	POP office Protocol	Presentation of POP	POP Office Presentation	None	POP office Protocol
27	Account information is accessed by using _____ command in FTP	ABORT	ACOV	ACCT	None	ACOV
28	the Compression method normally used in _____ encoding.	File Length	Byte Length	Run Length	None	Run Length
29	In Class A, First byte refers to _____	Netid	hosted	Mask	Subnet id	Netid
30	The Third level of hierarchy in the subnet is _____	Site id	Host id	Mask	Subnet id	Subnet id
31	_____ joins several Classes	Subnet	Supernet	Milnet	Arpranet	Subnet
32	The class of 208.34.55.12 is _____	Class A	Class B	Class C	Class D	Class C
33	The Net id of 114.34.2.8 IP Address is _____	114	114.34	114.34.2	14.43.28	114
34	The Value of Default Mask is _____	255.0.0.0	255.255.0.0	255.255.255.0	255.255.255.255	255.255.0.0
35	The Hardware deice that is used transfer data from one network to another is _____	Router	Bridge	Gateway	Repeater	Gateway
36	If the IP Address is 20.1.1.02 then the class of the address is _____	Class A	Class B	Class C	Class D	Class A
37	NTP refers to _____	Network Transfer Protocol	Network Traffic Protocol	Network Time Protocol	None	Network Time Protocol
38	PV6 refers to _____	IP Version Number	ARP Version Number	RARP Version Number	None.	IP Version Number
39	Two types of Subnetting are _____ and _____	Static, Variable	Dynamic, Static	Fixed, Static	None	Static, Variable
40	If the IP address is 123.12.3.2, then Host id is _____	3.2	12.3.2	12.3	None	3.2
41	The Network Class of 187.12.12.6 is _____	Class A	Class B	Class C	Class D	Class B
42	All ARP request packets are transmitted with the _____	Ethernet broadcast address	Ethernet Unicast address	Ethernet Multicast address	Both a and c	Ethernet broadcast address
43	ARP reply packets is directed to the _____	Host	Router	Bridge	None	Host
44	The size of an ARP request or reply packet is _____	24 Bytes	12 Bytes	6 Bytes	28 Bytes.	28 Bytes.
45	IP belongs to the _____ in the OSI model.	Network Layer	Session Layer	Transport Layer	Presentation Layer	Network Layer
46	ICMP Refers to _____	Internet Control Management Protocol	Internet Control Message Protocol	Internet Control Middleware Protocol	None	Internet Control Management Protocol
47	Private networks can provide ___for organizations	Efficiency	Privacy	A and B	None of the above	A and B
48	Both private and hybrid networks have a major drawback: _____	Lack of privacy	Cost	Lack of security	Time consuming	Cost
49	VPN is a network that is _____ but _____.	private; public	private; virtual	Public ; virtual	None of the above	private; virtual

50	VPN is physically _____ but virtually _____.	public; private	private; virtual	Public ; virtual	None of the above	public; private
51	An _____ is a private access that uses the TCP/IP protocol suite	Internet	Extranet	Intranet	Ethernet	Intranet
52	A _____ network is totally isolated from the global Internet	Private	Hybrid	Virtual	None	Private
53	A VPN can use _____ to guarantee privacy	IP Sec	Tunneling	Both a and b	None	Tunneling
54	On a network that uses NAT, the _____ has a translation table.	Bridge	Router	Server	None of above	Router
55	On a network that uses NAT, _____ initiates the communication	An internal host	An External host	Router	Hub	An internal host
56	On a network that uses NAT, the router can use _____ global address	1	2	3	None of the above	1
57	An IPv6 address is _____ bits long.	32	64	128	None	128
58	IPv6 allows _____ security provisions than IPv4.	More	Less	Same Level	None	More
59	An IPv6 address consists of _____ bytes	4	8	16	32	4
60	An IPv6 address can have up to _____ colons	8	7	4	None	8
61	_____ Address defines a group of computers.	Unicast	Multicast	Broadcast	None	Multicast

Reg.No ----------

[17CTU403]

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)

(Established Under Section 3 of UGC Act 1956)

COIMBATORE – 641 021

B.Sc. DEGREE EXAMINATION

Fourth Semester - I Internal Examination

December 2018

Computer Technology

Internetworking with TCP/IP

Time: 2 Hrs **Max.Marks: 50**

Date : 18-12-2018 **Class: II B.Sc (CT)**

Part - A **(20 X 1 = 20)**

Answer All the Questions

1. The TCP layers consist of _____layers.

   a) 6 b) 5 c) 7 d) 9

2. The ____layer is also called as IP layer.

   a) Application Layer b) data-link layer c) Network layer d) Physical Layer

3. The Network layer consists of ____sub-protocols.

   a) 4 b) 5 c) 3 d) 6

4. When a Physical address is given which of the following returns IP Address?

   a) ICMP b) IGMP c) ARQ d) RARQ

5. When an IP address is given which of the following returns physical Address?

   a) IGMP b) ICMP c) RARQ d) ARQ

6. Which of the following is not a switching element?

   a) LAN b) Switch c) HUB d) Router

7. _____ is a network which is created out of a main network.

   a) Supernet b) Subnet c) LAN d) WAN

8. When packets are send in Connection-less method it is denoted as _____.

   a) IP Datagram b) Datagram c) Packets d) None

9. When packets are send in Connection oriented method it is denoted as _____.

   a) Subnet b)supernet c) packet d) None

10. IP address cannot be represented in _____format.

    a) Hexadecimal b) Dotted Decimal c) Octal d) Binary

11. An IPv4 address is represented in ___ bits

    a) 32 b) 64 c) 128 d) 16

12. There are ____ types of classes in the classful addressing.

    a) 4 b) 5 c) 10 d) 6

13. ____ is a device used for connecting different networks.

a) Switch b) hub c) router d) Multiplexer

14. _____ refers to the number of systems that are not used in a network.

a) Hide b) Mask c) Multi d) No-operation

15. Which of the following IP belong to Class-A IP Address?

a) 12.12.12.12 b) 12.12.12.12.12 c) 171.14.23.1 d) None

16. Which of the following IP refers to Class B Addressing?

a) 172.16.25.1 b) 172.16.350.23 c) 12.12.12.12 d) None

17. The set of rules which are followed and made legal is called _____

a) De-Facto b) De-jurie c) Protocol d) All the above

18. Which of the following is used in Wireless communication?

a) Co-axial b) Twisted c) Fiber-optics d) None

19. Which of the following IP address is legal?

a) 172.16.25.1 b) 172.156.024.0 c) 256.0.0.12 d) None

20. The Top four layers of OSI Reference model is collectively known as ___ in TCP.

a) Physical layer b) Data-link layer c) Application Layer d) Network Layer

Part - B **(3 X 2 = 6)**

Answer all the Questions

21. What is a Network?

22. List the layers of OSI reference Model.

23. What is the use of ARP?

Part - C **(3 X 8 = 24)**

Answer all the Questions

24. a) Describe TCP/IP protocol suite in detail.

(OR)

b) Explain about Classless Addressing.

25. a) Discuss about Various Interconnecting Devices.

(OR)

b) Convert the following IP address into other Notations

i) 172.16.25.1 ii) 00001111 00001111 11110000 00001010

26. a) Write a detail note on Classful Addressing.

(OR)

b) Write a detail note on ARQ.

Reg.No ----------

[17CTU403]

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)

(Established Under Section 3 of UGC Act 1956)

COIMBATORE – 641 021

B.Sc. DEGREE EXAMINATION

Third Semester - I Internal Examination

December 2018

Computer Technology

Internetworking with TCP/IP

Time: 2 Hrs **Max.Marks: 50**

Date : 18-12-2018 **Class: II B.Sc (CT)**

Part - A **(20 X 1 = 20)**

Answer All the Questions

1. The TCP layers consist of _____layers.

   a) 6 b) **5** c) 7 d) 9

2. The ____layer is also called as IP layer.

   a) Application Layer b) data-link layer c) **Network layer** d) Physical Layer

3. The Network layer consists of ____sub-protocols.

   a) **4** b) 5 c) 3 d) 6

4. When a Physical address is given which of the following returns IP Address?

   a) ICMP b) IGMP c) ARQ d) **RARP**

5. When an IP address is given which of the following returns physical Address?

   a) IGMP b) ICMP c) RARQ d) **ARP**

6. Which of the following is not a switching element?

   a) **LAN** b) Switch c) HUB d) Router

7. _____ is a network which is created out of a main network.

   a) Supernet b) **Subnet** c) LAN d) WAN

8. When packets are send in Connection-less method it is denoted as _____.

   a) IP Datagram b) **Datagram** c) Packets d) None

9. When packets are send in Connection oriented method it is denoted as _____.

   a) Subnet b)supernet c) **packet** d) None

10. IP address cannot be represented in _____format.

    a) Hexadecimal b) Dotted Decimal c) **Octal** d) Binary

11. An IPv4 address is represented in ___ bits

    a) **32** b) 64 c) 128 d) 16

12. There are ____ types of classes in the classful addressing.

    a) 4 b) **5** c) 10 d) 6

13. ____ is a device used for connecting different networks.

a) Switch b) hub c) **router** d) Multiplexer

14. _____ refers to the number of systems that are not used in a network.

a) Hide b) **Mask** c) Multi d) No-operation

15. Which of the following IP belong to Class-A IP Address?

a) **12.12.12.12** b) 12.12.12.12.12 c) 171.14.23.1 d) None

16. Which of the following IP refers to Class B Addressing?

a) **172.16.25.1** b) 172.16.350.23 c) 12.12.12.12 d) None

17. The set of rules which are followed and made legal is called _____

a) De-Facto b) **De-jurie** c) Protocol d) All the above

18. Which of the following is used in Wireless communication?

a) Co-axial b) Twisted c) Fiber-optics d) **None**

19. Which of the following IP address is legal?

a) **172.16.25.1** b) 172.156.024.0 c) 256.0.0.12 d) None

20. The Top four layers of OSI Reference model is collectively known as ___ in TCP.

a) Physical layer b) Data-link layer c) **Application Layer** d) Network Layer

Part - B **(3 X 2 = 6)**

Answer all the Questions

21. What is a Network?

Ans: Collection of devices interconnected using communication-link is known as Network

22. List the layers of OSI reference Model.

Ans: Physical Layer, Data-link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer & Application Layer.

23. What is the use of ARP?

Ans: A protocol which retrieves the Physical address when IP-Address is provided.

Part - C **(3 X 8 = 24)**

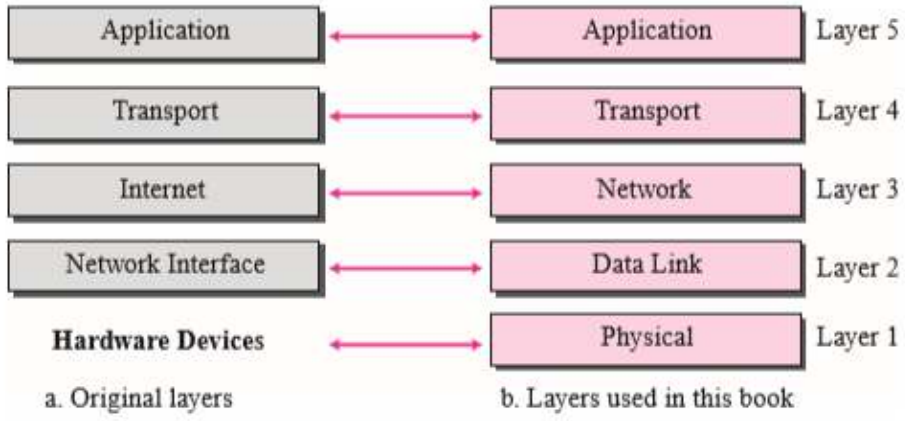
Answer all the Questions

24. a) Describe TCP/IP protocol suite in detail.

Ans:

The TCP/IP protocol suite was developed prior to the OSI model. The original TCP/IP protocol suite was defined as four software layers built upon the hardware. Today, however, TCP/IP is thought of as a five-layer model with the layers named similarly to the ones in the OSI model.

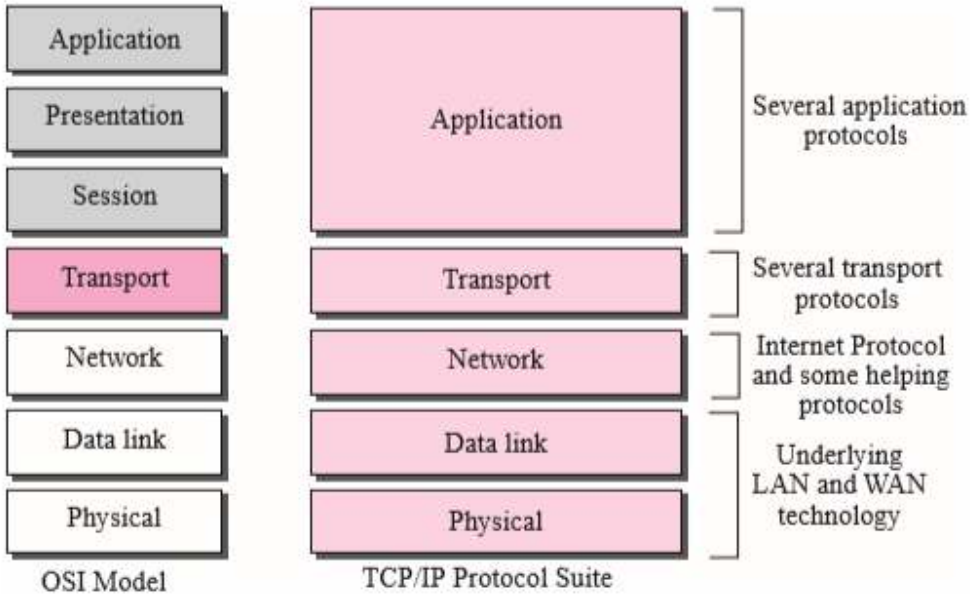
Figure *Layers in the TCP/IP Protocol Suite*



Comparison between OSI and TCP/IP Protocol Suite

Two reasons were mentioned for this decision. First, TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport layer protocols. Second, the application layer is not only one piece of software. Many applications can be developed at this layer. If some of the functionalities mentioned in the session and presentation are needed for a particular application, it can be included in the development of that piece of software. TCP/IP is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality, but the modules are not necessarily interdependent. Whereas the OSI model specifies which functions belong to each of its layers, the layers of the TCP/IP protocol suite contain relatively independent protocols that can be mixed and matched, depending on the needs of the system.

Figure *TCP/IP and OSI model*



Physical Link Layer

TCP/IP does not define any specific protocol for the physical layer. It supports all of the standard and proprietary protocols. At this level, the communication is between two hops or nodes, either a computer or router. The unit of communication is a single bit. When the connection is established between the two nodes, a stream of bits is flowing between them. The physical layer, however, treats each bit individually. We are assuming that at this moment the two computers have discovered that the most efficient way to communicate with each other is via routers.

Note that if a node is connected to n links, it needs n physical-layer protocols, one for each link. The reason is that different links may use different physical-layer protocols. Each computer involves with only one link; each router involves with only two links.

The responsibility of the physical layer, in addition to delivery of bits, matches with what mentioned for the physical layer of the OSI model, but it mostly depends on the underlying technologies that provide links.

Data Link Layer

TCP/IP does not define any specific protocol for the data link layer either. It supports all of the standard and proprietary protocols. At this level, the communication is also between two hops or nodes. The unit of communication however, is a packet called a frame. A frame is a packet that encapsulates the data received from the network layer with an added header and sometimes a trailer. The head, among other communication information, includes the source and destination of frame. The destination address is needed to define the right recipient of the frame because many nodes may have been connected to the link. The source address is needed for possible response or acknowledgment as may be required by some protocols.

Network Layer

At the network layer (or, more accurately, the internetwork layer), TCP/IP supports the Internet Protocol (IP). The Internet Protocol (IP) is the transmission mechanism used by the TCP/IP protocols. IP transports data in packets called datagrams, each of which is transported separately. Datagrams can travel along different routes and can arrive out of sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination.

Note that there is a main difference between the communication at the network layer and the communication at data link or physical layers. Communication at the network layer is end to end while the communications at the other two layers are node to node.

Transport Layer

There is a main difference between the transport layer and the network layer. Although all nodes in a network need to have the network layer, only the two end computers need to have the transport layer. The network layer is responsible for sending individual datagrams from computer A to computer B; the transport layer is responsible for delivering the whole message, which is called a segment, a user datagram, or a packet, from A to B. A segment may consist of a few or tens of datagrams. The segments need to be broken into datagrams and each datagram has to be delivered to the network layer for transmission. Since the Internet defines a different route for each datagram, the datagrams may arrive out of order and may be lost. The transport layer at computer B needs to wait until all of these datagrams to arrive, assemble them and make a segment out of them.

Again, we should know that the two transport layers only think that they are communicating with each other using a segment; the communication is done through the physical layer and the exchange of bits. Traditionally, the transport layer was represented in the TCP/IP suite by two protocols: User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). A new protocol called Stream Control Transmission Protocol (SCTP) has been introduced in the last few years.

Application Layer

The application layer in TCP/IP is equivalent to the combined session, presentation, and application layers in the OSI model. The application layer allows a user to access the services of our private internet or the global Internet. Many protocols are defined at this layer to provide services such as electronic mail, file transfer, accessing the World Wide Web, and so on. Note that the communication at the application layer, like the one at the transport layer, is end to end. A message generated at computer A is sent to computer B without being changed during the transmission.

(OR)

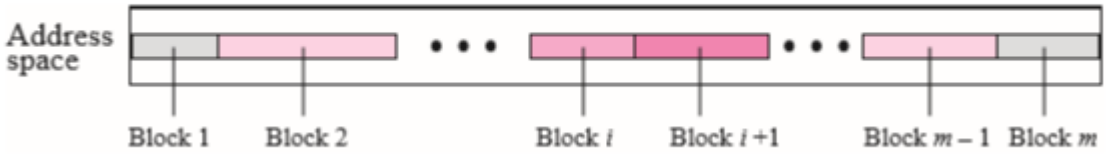
b) Explain about Classless Addressing.

Ans: Subnetting and supernetting in classful addressing did not really solve the address depletion problem and made the distribution of addresses and the routing process more difficult. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses to be increased, which means the format of the IP packets needs to be changed. The short-term solution still used in IPv4 addresses is called classless addressing. In other words, the class privilege was removed from the distribution to compensate for the address depletion.

Variable-Length Blocks

In classless addressing, the whole address space is divided into variable length blocks. Theoretically, we can have a block of 2 0, 2 1, 2 2. . . 2 32 addresses. The only restriction is that the number of addresses in a block needs to be a power of 2. An organization can be granted one block of addresses. The following figure shows the division of the whole address space into non-overlapping blocks.

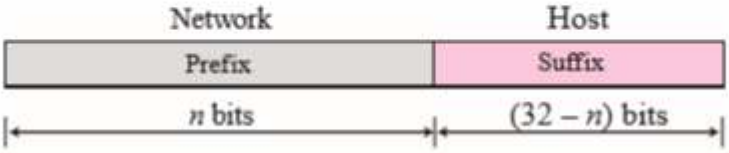
Figure *Variable-length blocks in classless addressing*



Two-Level Addressing

In classful addressing, two-level addressing was provided by dividing an address into netid and hostid. The same idea can be applied in classless addressing. When an organization is granted a block of addresses, the block is actually divided into two parts, the prefix and the suffix. The prefix plays the same role as the netid; the suffix plays the same role as the hostid. All addresses in the block have the same prefix; each address has a different suffix. The following figure shows the prefix and suffix in a classless block.

Figure *Prefix and suffix*



In classful addressing, the length of the netid, n, depends on the class of the address; it can be only 8, 16, or 24. In classless addressing, the length of the prefix, n, depends on the size of the block; it can be 0, 1, 2, 3, . . . , 32. In classless addressing, the value of n is referred to as prefix length; the value of 32 − n is referred to as suffix length.

Slash Notation

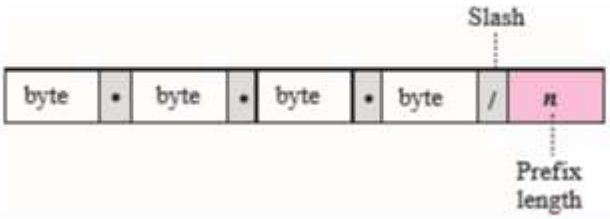
The netid length in classful addressing or the prefix length in classless addressing play a very important role when we need to extract the information about the block from a given address in the block. However, there is a difference here in classful and classless addressing.

- In classful addressing, the netid length is inherent in the address. Given an address, we know the class of the address that allows us to find the netid length (8, 16, or 24).

- In classless addressing, the prefix length cannot be found if we are given only an address in the block. The given address can belong to a block with any prefix length.
- In classless addressing, we need to include the prefix length to each address if we need to find the block of the address.

In this case, the prefix length, n, is added to the address separated by a slash. The notation is informally referred to as slash notation. An address in classless addressing can then be represented as shown below

Figure *Slash notation*



The slash notation is formally referred to as classless interdomain routing or CIDR (pronounced cider) notation.

Extracting Block Information

An address in slash notation (CIDR) contains all information we need about the block: the first address (network address), the number of addresses, and the last address. These three pieces of information can be found as follows:

- The number of addresses in the block can be found as:

  $$N = 232 - n$$

  in which n is the prefix length and N is the number of addresses in the block.
- The first address (network address) in the block can be found by ANDing the address with the network mask:

  First address = (any address) AND (network mask)

  Alternatively, we can keep the n leftmost bits of any address in the block and set the 32− n bits to 0s to find the first address.
- The last address in the block can be found by either adding the first address with the number of addresses or, directly, by ORing the address with the complement (NOTing) of the network mask:

  Last address = (any address) OR [NOT (network mask)]

  Alternatively, we can keep the n leftmost bits of any address in the block and set the 32 − n bits to 1s to find the last address.

Block Allocation

The next issue in classless addressing is block allocation. How are the blocks allocated? The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Addresses (ICANN). However, ICANN does not normally allocate addresses to individual Internet users. It assigns a large block of addresses to an ISP (or a larger organization that is considered an ISP in this case). For the proper operation of the CIDR, three restrictions need to be applied to the allocated block. 1. The number of requested addresses, N, needs to be a power of 2. This is needed to provide an integer value for the prefix length, n (see the second restriction). The number of addresses can be 1, 2, 4, 8, 16, and so on. 2. The value of prefix length can be found from the number of addresses in the block. Since $N = 232 - n$, then $n = \log2\ (232/N) = 32 - \log2N$. That is the reason why N needs to be a power of 2. 3. The requested block needs to be allocated where there are a contiguous number of unallocated addresses in the address space. However, there is a restriction on choosing the beginning addresses of the block. The beginning address needs to be divisible by the number of addresses in the block. To see this restriction, we can show that the beginning address can be calculated as $X \times 2n - 32$ in which X is the decimal value of the prefix. In other words, the beginning address is $X \times N$.

Subnetting

Three levels of hierarchy can be created using subnetting. An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a subnetwork (or subnet). The concept is the same as we discussed for classful addressing. Note that nothing stops the organization from creating more levels. A subnetwork can be divided into several sub-subnetworks. A sub-subnetwork can be divided into several sub-sub-subnetworks. And so on.

Designing Subnets

The subnetworks in a network should be carefully designed to enable the routing of packets. We assume the total number of addresses granted to the organization is N, the prefix length is n, the assigned number of addresses to each subnetwork is Nsub, the prefix length for each subnetwork is nsub, and the total number of subnetworks is s. Then, the following steps need to be carefully followed to guarantee the proper operation of the subnetworks.

1. The number of addresses in each subnetwork should be a power of 2.
2. The prefix length for each subnetwork should be found using the following formula:

$$nsub = n + \log2\ (N/Nsub)$$

3. The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork. This can be achieved if we first assign addresses to larger networks.

Address Aggregation

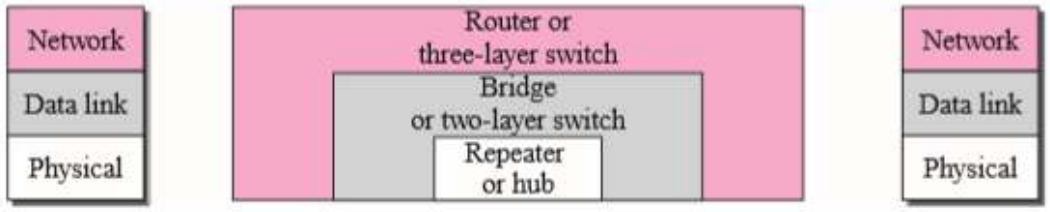
One of the advantages of CIDR architecture is address aggregation. ICANN assigns a large block of addresses to an ISP. Each ISP in turn divides its assigned block into smaller subblocks and grants the subblocks to its customers; many blocks of addresses are aggregated in one block and granted to one ISP.

25. a) Discuss about Various Interconnecting Devices.

Ans: LANs or WANs do not normally operate in isolation. They are connected to one another or to the Internet. To connect LANs and WANs together we use connecting devices. Connecting devices can operate in different layers of the Internet model.

We discuss three kinds of connecting devices: repeaters (or hubs), bridges (or two-layer switches), and routers (or three-layer switches). Repeaters and hubs operate in the first layer of the Internet model. Bridges and two-layer switches operate in the first two layers. Routers and three-layer switches operate in the first three layers. Figure shows the layers in which each device operates.

Figure *Connecting devices*



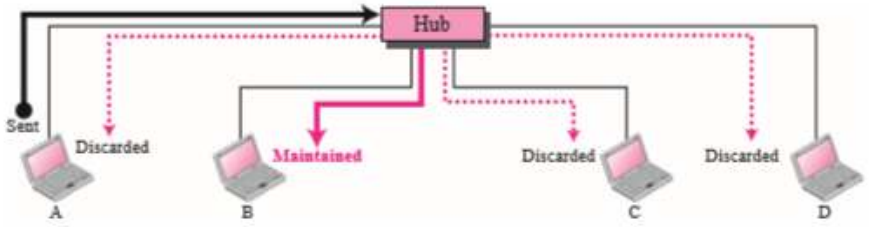
i) Repeaters

A repeater is a device that operates only in the physical layer. Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data. A repeater receives a signal and, before it becomes too weak or corrupted, regenerates and retimes the original bit pattern. The repeater then sends the refreshed signal. In the past, when Ethernet LANs were using bus topology, a repeater was used to connect two segments of a LAN to overcome the length restriction of the coaxial cable. Today, however, Ethernet LANs use star topology. In a star topology, a repeater is a multiport device, often called a hub that can be used to serve as the connecting point and at the same time function as a repeater.

The following figure shows that when a packet from station A to B arrives at the hub, the signal representing the frame is regenerated to remove any possible corrupting noise, but

the hub forwards the packet from all outgoing port to all stations in the LAN. In other words, the frame is broadcast. All stations in the LAN receive the frame, but only station B keeps it. The rest of the stations discard it. Figure also shows the role of a repeater or a hub in a switched LAN. The figure definitely shows that a hub does not have a filtering capability; it does not have the intelligence to find from which port the frame should be sent out.

Figure 3.41 *Repeater or hub*



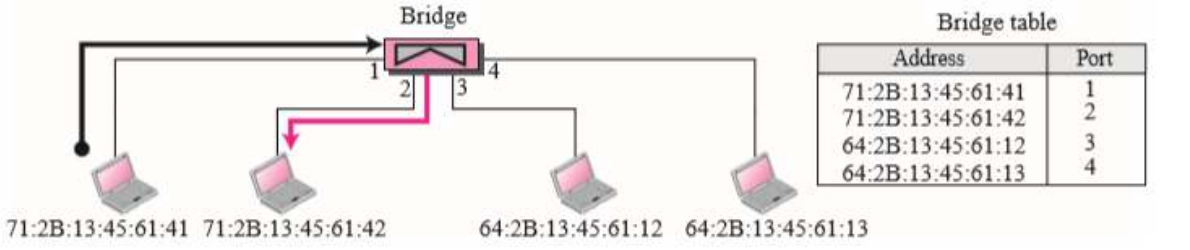
A hub or a repeater is a physical-layer device. They do not have any data-link address and they do not check the data-link address of the received frame. They just regenerate the corrupted bits and send them out from every port.

ii) Bridges

A bridge operates in both the physical and the data link layers. As a physical-layer device, it regenerates the signal it receives. As a data link layer device, the bridge can check the MAC addresses (source and destination) contained in the frame.

Filtering: A bridge has filtering capability. It can check the destination address of a frame and can decide from which outgoing port the frame should be send out. Let us give an example. The following figure, we have a LAN with four stations that are connected to a bridge. If a frame destined for station 71:2B:13:45:61:42 arrives at port 1, the bridge consults its table to find the departing port. According to its table, frames for 71:2B:13:45:61:42 should be sent out only through port 2; therefore, there is no need for forwarding the frame through other ports

Figure *Bridge*



Bridge table

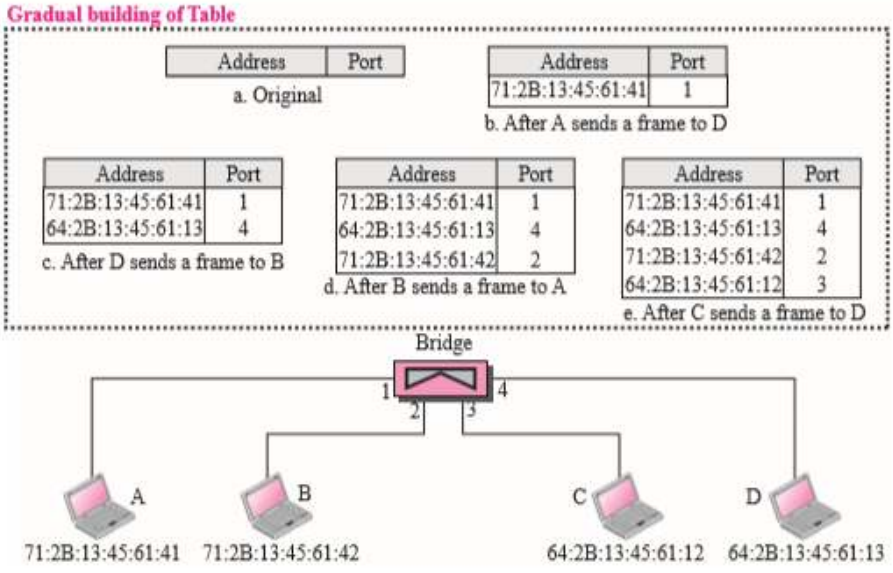
Address	Port
71:2B:13:45:61:41	1
71:2B:13:45:61:42	2
64:2B:13:45:61:12	3
64:2B:13:45:61:13	4

Transparent Bridges: A transparent bridge is a bridge in which the stations are completely unaware of the bridge's existence. If a bridge is added or deleted from the system, reconfiguration of the stations is unnecessary. According to the IEEE 802.1d specification, a system equipped with transparent bridges must meet three criteria: 1. Frames must be forwarded from one station to another. 2. The forwarding table is automatically made by learning frame movements in the network. 3. Loops in the system must be prevented.

Forwarding: A transparent bridge must correctly forward the frames, as discussed in the previous section.

Learning: The earliest bridges had forwarding tables that were static. The system administrator would manually enter each table entry during bridge setup. Although the process was simple, it was not practical. If a station was added or deleted, the table had to be modified manually. The same was true if a station's MAC address changed, which is not a rare event. For example, putting in a new network card means a new MAC address. A better solution to the static table is a dynamic table that maps addresses to ports automatically. To make a table dynamic, we need a bridge that gradually learns from the frame movements. To do this, the bridge inspects both the destination and the source addresses. The destination address is used for the forwarding decision (table lookup); the source address is used for adding entries to the table and for updating purposes. Let us elaborate on this process using Figure

Figure *Learning bridge*



1. When station A sends a frame to station D, the bridge does not have an entry for either D or A. The frame goes out from all three ports; the frame floods the network. However, by looking at the source address, the bridge learns that station A must be

connected to port 1. This means that frames destined for A, in the future, must be sent out through port 1. The bridge adds this entry to its table. The table has its first entry now.

2. When station D sends a frame to station B, the bridge has no entry for B, so it floods the network again. However, it adds one more entry to the table.

3. The learning process continues until the table has information about every port.

Two-Layer Switch: When we use the term switch, we must be careful because a switch can mean two different things. We must clarify the term by adding the level at which the device operates. We can have a two-layer switch or a three-layer switch. A two-layer switch performs at the physical and data link layer; it is a sophisticated bridge with faster forwarding capability.

iii) Routers

A router is a three-layer device; it operates in the physical, data link, and network layers. As a physical layer device, it regenerates the signal it receives. As a data link layer device, the router checks the physical addresses (source and destination) contained in the packet. As a network layer device, a router checks the network layer addresses (addresses in the IP layer). Note that bridges change collision domains, but routers limit broadcast domains.

A router can connect LANs together; a router can connect WANs together; and a router can connect LANs and WANs together. In other words, a router is an internetworking device; it connects independent networks together to form an internetwork. According to this definition, two networks (LANs or WANs) connected by a router become an internetwork or an internet.

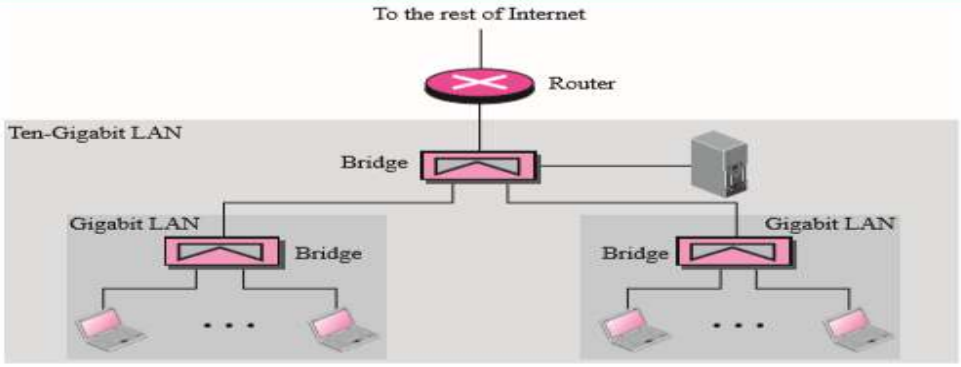
There are three major differences between a router and a repeater or a bridge.

1. A router has a physical and logical (IP) address for each of its interfaces.
2. A router acts only on those packets in which the physical destination address matches the address of the interface at which the packet arrives.
3. A router changes the physical address of the packet (both source and destination) when it forwards the packet.

Let us give an example. In the following Figure we assume an organization has two separate buildings with a Gigabit Ethernet LANs installed in each building. The organization uses bridges in each LAN. The two LANs can be connected together to form a larger LAN using Ten-Gigabit Ethernet technology that speeds up the connection to the Ethernet and the connection to the organization server. A router then can connect the whole system to the

Internet. A router will change the MAC address it receives because the MAC addresses have only local jurisdictions.

Figure 3.44 *Routing example*



Three-Layer Switch: A three-layer switch is a router; a router with an improved design to allow better performance. A three-layer switch can receive, process, and dispatch a packet much faster than a traditional router even though the functionality is the same.

(OR)

b) Convert the following IP address into other Notations

Ans:

i) 172.16.25.1

172 - 1010 1100

16 - 0001 0000

25 - 0001 1001

1 - 0000 0001

Binary Notation: $(10101100\ 00010000\ 00011001\ 00000001)_2$

Hexadecimal Notation: 0X AC101901

ii) 00001111 00001111 11110000 00001010

00001111 - 15

00001111 - 15

11110000 - 240

00001010 - 10

Dotted Decimal Notation: 15.15.240.10

Hexadecimal Notation:

0000 - 0 1111 - F 0000 - 0 1111 - F 1111 -F 0000 - 0 0000 - 0 1010 - A

0X 0F0FF00A

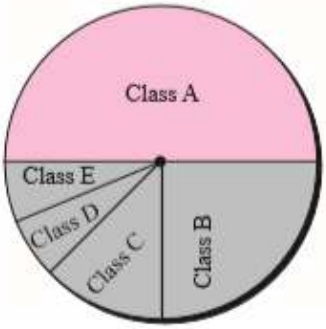
26. a) Write a detail note on Classful Addressing.

Ans:

IP addresses, when started a few decades ago, used the concept of classes. This architecture is called classful addressing.

Classes: In classful addressing, the IP address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the whole address space. The following Figure shows the class occupation of the address space.

Figure *Occupation of the address space*



Class A: 2^{31} = 2,147,483,648 addresses, 50%

Class B: 2^{30} = 1,073,741,824 addresses, 25%

Class C: 2^{29} = 536,870,912 addresses, 12.5%

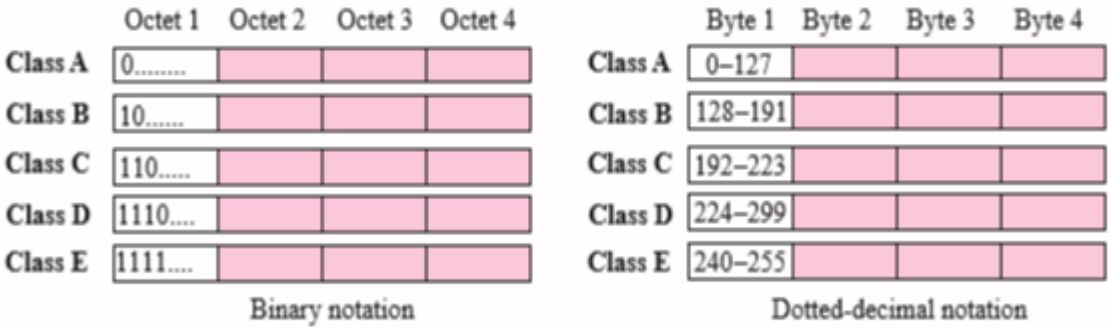
Class D: 2^{28} = 268,435,456 addresses, 6.25%

Class E: 2^{28} = 268,435,456 addresses, 6.25%

Recognizing Classes

We can find the class of an address when the address is given either in binary or in dotted decimal notation. In the binary notation, the first few bits can immediately tell us the class of the address; in the dotted-decimal notation, the value of the first byte can give the class of an address.

Figure *Finding the class of an address*



Note that some special addresses fall in class A or E. We emphasize that these special addresses are exceptions to the classification;

Ex:

Find the class of each address:

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 10100111 11011011 10001011 01101111

d. 11110011 10011011 11111011 00001111

Answer

a. The first bit is 0. This is a class A address.

b. The first 2 bits are 1; the third bit is 0. This is a class C address.

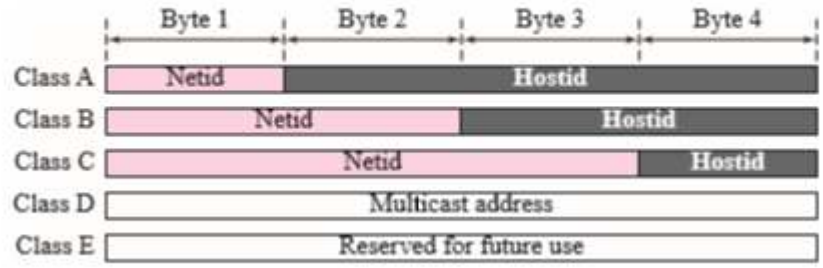
c. The first bit is 1; the second bit is 0. This is a class B address.

d. The first 4 bits are 1s. This is a class E address.

Netid and Hostid

In classful addressing, an IP address in classes A, B, and C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address. The following figure shows the netid and hostid bytes. Note that classes D and E are not divided into netid and hosted. In class A, 1 byte defines the netid and 3 bytes define the hostid. In class B, 2 bytes define the netid and 2 bytes define the hostid. In class C, 3 bytes define the netid and 1 byte defines the hostid.

Figure *Netid and hostid*



Classes and Blocks: One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size. Let us look at each class.

Class A: Since only 1 byte in class A defines the netid and the leftmost bit should be 0, the next 7 bits can be changed to find the number of blocks in this class. Therefore, class A is divided into 27 = 128 blocks that can be assigned to 128 organizations (the number is less because some blocks were reserved as special blocks). However, each block in this class contains 16,777,216 addresses, which means the organization should be a really large one to use all these addresses. Many addresses are wasted in this class. Figure 5.9 shows the block in class A.

Class B: Since 2 bytes in class B define the class and the two leftmost bit should be 10 (fixed), the next 14 bits can be changed to find the number of blocks in this class. Therefore, class B is divided into 214 = 16,384 blocks that can be assigned to 16,384 organizations (the number is less because some blocks were reserved as special blocks). However, each block in

this class contains 65,536 addresses. Not so many organizations can use so many addresses. Many addresses are wasted in this class. Figure 5.10 shows the blocks in class B.

Class C: Since 3 bytes in class C define the class and the three leftmost bits should be 110 (fixed), the next 21 bits can be changed to find the number of blocks in this class. Therefore, class C is divided into 221 = 2,097,152 blocks, in which each block contains 256 addresses that can be assigned to 2,097,152 organizations (the number is less because some blocks were reserved as special blocks). Each block contains 256 addresses. However, not so many organizations were so small as to be satisfied with a class C block. Figure 5.11 shows the blocks in class C.

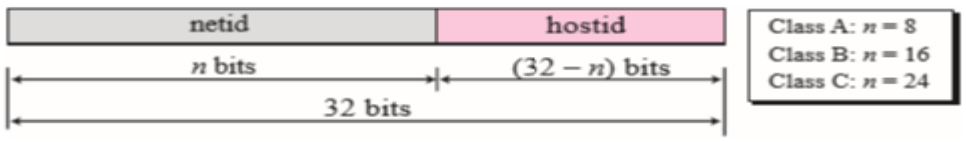
Class D: There is just one block of class D addresses. It is designed for multicasting, as we will see in a later section. Each address in this class is used to define one group of hosts on the Internet. When a group is assigned an address in this class, every host that is a member of this group will have a multicast address in addition to its normal (unicast) address.

Class E: There is just one block of class E addresses. It was designed for use as reserved addresses.

Two-Level Addressing

When classful addressing was designed, it was assumed that the whole Internet is divided into many networks and each network connects many hosts. In other words, the Internet was seen as a network of networks. A network was normally created by an organization that wanted to be connected to the Internet. The Internet authorities allocated a block of addresses to the organization (in class A, B, or C). Since all addresses in a network belonged to a single block, each address in classful addressing contains two parts: netid and hostid. The netid defines the network; the hostid defines a particular host connected to that network. If n bits in the class defines the net, then 32 − n bits defines the host. However, the value of n depends on the class the block belongs to. The value of n can be 8, 16 or 24 corresponding to classes A, B, and C respectively.

Figure *Two-level addressing in classful addressing*

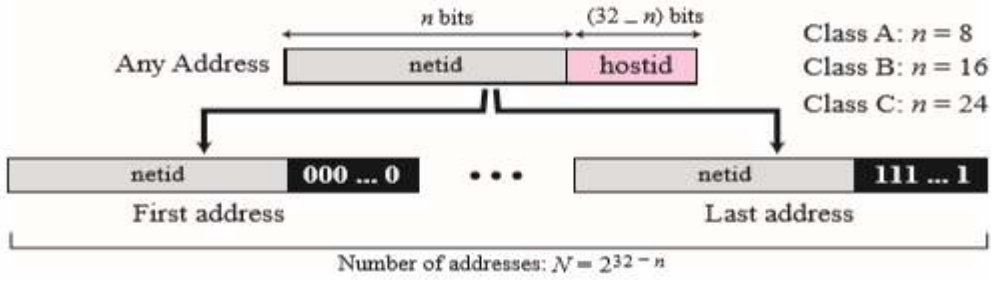


Extracting Information in a Block:

A block is a range of addresses. Given any address in the block, we normally like to know three pieces of information about the block: the number of addresses, the first address, and the last address. Before we can extract these pieces of information, we need to know the

class of the address, which we showed how to find in the previous section. After the class of the block is found, we know the value of n, the length of netid in bits.

Figure *Information extraction in classful addressing*



We can now find these three pieces of information in the above figure.

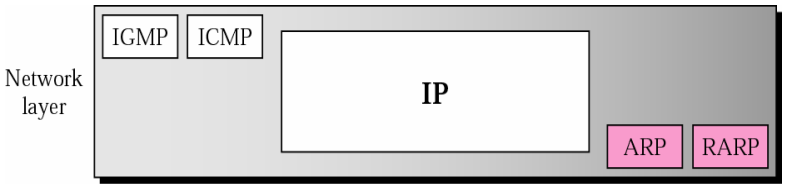
- The number of addresses in the block, N, can be found using N = 232−n.
- To find the first address, we keep the n leftmost bits and set the (32 − n) rightmost bits all to 0s.
- To find the last address, we keep the n leftmost bits and set the (32 − n) rightmost bits all to 1s.

(OR)

b) Write a detail note on ARQ.

Ans:

Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. A mapping corresponds a logical address to a physical address.



Position of ARP in TCP/IP Suite

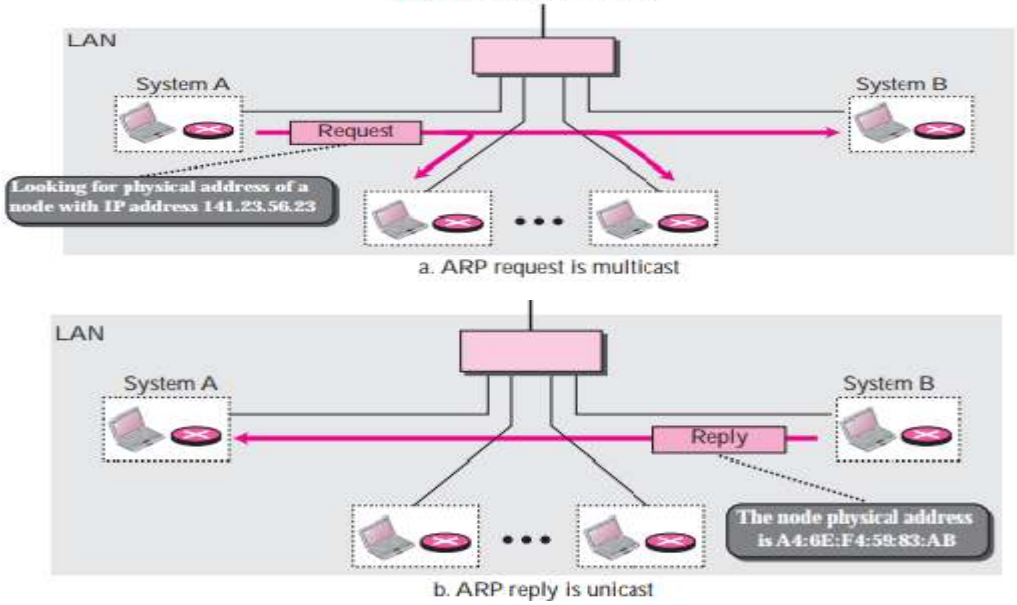
ARP associates an IP address with its physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address that is usually imprinted on the NIC.

Anytime a host, or a router, needs to find the physical address of another host or router on its network, it sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network.

Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and physical addresses. The packet is unicast directly to the inquirer using the physical address received in the query packet.

In the following figure, the system (A) has a packet that needs to be delivered to another system (B) with IP address 141.23.56.23. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of 141.23.56.23. This packet is received by every system on the physical network, but only system-B will reply with ARP-Replay. After receiving the reply the system (A) now uses the physical address to communicate with System-B.

Figure *ARP operation*



Packet Format

The Packet format for ARP is given below.

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

The fields are as follows

Hardware type (HTYPE): This field specifies the Link Layer protocol type. Example: Ethernet is 1.

Protocol type (PTYPE): This field specifies the upper layer protocol for which the ARP request is intended. For example, Internet Protocol (IPv4) is encoded as 0x0800.

Hardware length (HLEN): Length (in octets) of a hardware address. Ethernet addresses size is 6.

Protocol length (PLEN): Length (in octets) of a logical address of the specified protocol (cf. PTYPE). IPv4 address size is 4.

Operation: Specifies the operation that the sender is performing: 1 for request, 2 for reply.

Sender hardware addresses (SHA): Hardware (MAC) address of the sender.

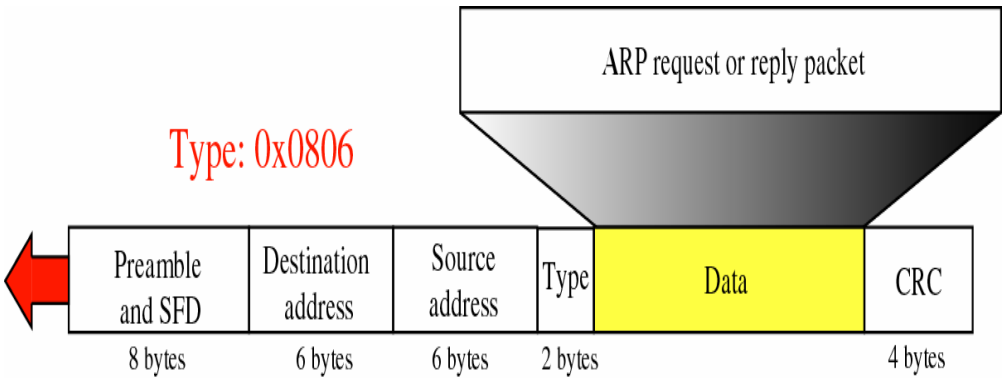
Sender protocols address (SPA): Upper layer protocol address of the sender.

Target hardware address (THA) : Hardware address of the intended receiver. This field is ignored in requests.

Target protocol address (TPA): Upper layer protocol address of the intended receiver.

Encapsulation

ARP packet is encapsulated directly into a data link frame ARP packet encapsulated in an Ethernet frame



Cases in which ARP is used:

Case 1: The sender is a host and wants to send a packet to another host on the same network. In this case, the logical address that must be mapped to a physical address is the destination IP address in the datagram header.

Case 2: The sender is a host and wants to send a packet to another host on another network. In this case, the host looks at its routing table and finds the IP address of the next hop (router) for this destination. If it does not have a routing table, it looks for the IP address of the default router. The IP address of the router becomes the logical address that must be mapped to a physical address.

Case 3: The sender is a router that has received a datagram destined for a host on another network. It checks its routing table and finds the IP address of the next router. The IP address of the next router becomes the logical address that must be mapped to a physical address.

Case 4: The sender is a router that has received a datagram destined for a host in the same network. The destination IP address of the datagram becomes the logical address that must be mapped to a physical address.

Reg.No ----------

[17CTU403]

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)

(Established Under Section 3 of UGC Act 1956)

COIMBATORE – 641 021

B.Sc. DEGREE EXAMINATION

Fourth Semester - II Internal Examination

February - 2018

Computer Technology

Internetworking with TCP/IP

Time: 2 Hrs **Max.Marks: 50**

Date & Session: 05-02-19 (FN) **Class: II B.Sc (CT)**

Part - A **(20 X 1 = 20)**

Answer All the Questions

1. ICMP is used to ____errors.

   a) Detect b) Correct c) Report d) All the above

2. Which of the following is the Error Reporting Protocol?

   a) ICMP b) IGMP c) ARQ d) RARQ

3. There are ___types of errors that is reported in ICMP.

   a) 4 b) 5 c) 3 d) 2

4. The minimum header size of IP Datagram is _____bytes

   a) 20 b) 40 c) 60 d) 65,535

5. The Maximum header size of IP Datagram is _____bytes

   a) 20 b) 40 c) 60 d) 65,535

6. The Size of the datagram is ____ bytes

   a) 65,535 b) 65,536 c) 32,765 d) 32,766

7. The process of dividing datagram into sub-datagram is called _____

   a) Segmentation b) Fragmentation c) decapsulation d) None

8. The packet in the Network Layer is called _____

   a) Packet b) Frames c) Process d) Datagram

9. Which of the following is a Group Management Protocol?

   a) ICMP b) IGMP c) ARQ d) RARQ

10. Which of the following is not a protocol of Network Layer?

    a) ICMP b) IP c) ARP d) TCP

11. The client Port is called as _____ port.

    a) Ephemeral b) Reserved c) Well-Known d) None

12. ICMP messages are encapsulated in the _____

    a) Frame b) Datagram c) Process d) Port

13. The Destination-Unreachable in ICMP error Message is of Type value ___

a) 3 b) 4 c) 5 d) 6

14. ____ is a command to check if a host exists or not?

a) ping b) isalive c) alive() d) netstat

15. The Top four layers of OSI Reference model is collectively known as ___ in TCP.

a) Physical layer b) Data-link layer c) Application Layer d) Network Layer

16. The TCP layers consist of _____layers.

a) 6 b) 5 c) 7 d) 9

17. Process to process communication is done through ____

a) HUB b) SWITCH c) PORT d) REPEATER

18. The OSI Reference Model Consists of ___ layers

a) 6 b) 5 c) 7 d) 9

19. Which of the following protocol is Connection Oriented?

a) UDP b) IP c) TCP d) TCP/IP

20. Process to Process communication is done in _____layer.

a) Network b) Data-link c) Physical d) Transport

Part - B **(3 X 2 = 6)**

Answer all the Questions

21. What is a Checksum?

22. Define a process in regards to Transport Layer.

23. What is ICMP?

Part - C **(3 X 8 = 24)**

Answer all the Questions

24. a) Explain about ICMP.

(OR)

b) Discuss about IP Options in detail.

25. a) What is a Port? Explain the types of ports in detail.

(OR)

b) Write a detail note on IP Fragmentation.

26. a) Discuss the need of IGMP in detail with the protocol format.

(OR)

b) Write in detail about IP Datagram Header.

17CTU403]

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)

(Established Under Section 3 of UGC Act 1956)

COIMBATORE – 641 021

B.Sc. DEGREE EXAMINATION

Fourth Semester - II Internal Examination

January 2018

Computer Technology

Internetworking with TCP/IP

Time: 2 Hrs **Max.Marks: 50**

Date : **Class: II B.Sc (CT)**

Part - A **(20 X 1 = 20)**

Answer All the Questions

1. ICMP is used to ____errors.

   a) Detect b) Correct c) **Report** d) All the above

2. Which of the following is the Error Reporting Protocol?

   a) **ICMP** b) IGMP c) ARQ d) RARQ

3. There are ___types of errors that is reported in ICMP.

   a) 4 b) 5 c) **3** d) 2

4. The minimum header size of IP Datagram is _____bytes

   a) **20** b) 40 c) 60 d) 65,535

5. The Maximum header size of IP Datagram is _____bytes

   a) 20 b) 40 c) **60** d) 65,535

6. The Size of the datagram is ____ bytes

   a) **65,535** b) 65,536 c) 32,765 d) 32,766

7. The process of dividing datagram into sub-datagram is called _____

   a) Segmentation b) **Fragmentation** c) decapsulation d) None

8. The packet in the Network Layer is called _____

   a) **Packet** b) Frames c) Process d) Datagram

9. Which of the following is a Group Management Protocol?

   a) ICMP b) **IGMP** c) ARQ d) RARQ

10. Which of the following is not a protocol of Network Layer?

    a) ICMP b) IP c) ARP d) **TCP**

11. The client Port is called as _____ port.

    a) **Ephemeral** b) Reserved c) Well-Known d) None

12. ICMP messages are encapsulated in the _____

    a) **Frame** b) Datagram c) Process d) Port

13. The Destination-Unreachable in ICMP error Message is of Type value ___

    a) 3 b) 4 c) 5 d) **6**

14. ____ is a command to check if a host exists or not?

    a) **ping** b) isalive c) alive() d) netstat

15. The Top four layers of OSI Reference model is collectively known as ___ in TCP.

    a) Physical layer b) Data-link layer c) **Application Layer** d) Network Layer

16. The TCP layers consist of _____layers.

    a) 6 b) **5** c) 7 d) 9

17. Process to process communication is done through ____

    a) HUB b) SWITCH c) **PORT** d) REPEATER

18. The OSI Reference Model Consists of ___ layers

    a) 6 b) 5 c) **7** d) 9

19. Which of the following protocol is Connection Oriented?

    a) UDP b) IP c) **TCP** d) TCP/IP

20. Process to Process communication is done in _____layer.

    a) Network b) Data-link c) Physical d) **Transport**

Part - B **(3 X 2 = 6)**

Answer all the Questions

21. What is a Checksum?

    Ans: A Calculation done in both end of the network to check the packet corruption.

22. Define a process in regards to Transport Layer.

    Ans: A running Program is called a Process

23. What is ICMP?

    Ans: It is a error reporting Protocol that resides in Network Layer of TCP/IP Protocol Suite.

Part - C **(3 X 8 = 24)**

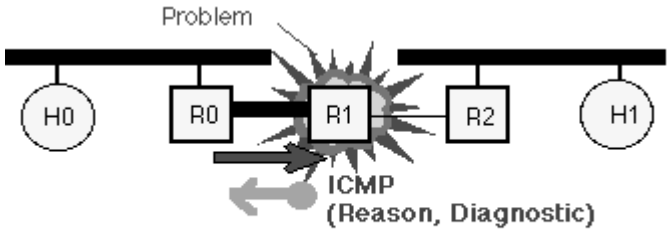
Answer all the Questions

24. a) Explain about ICMP.

    Ans: The Internet Control Message Protocol (ICMP) protocol is classic example of a client server application. The ICMP server executes on all IP end system computers and all IP intermediate systems (i.e routers). The protocol is used to report problems with delivery of IP datagrams within an IP network. It can be sued to show when a particular End System (ES) is not responding, when an IP network is not reachable, when a node is overloaded, when an error occurs in the IP header information, etc. The protocol is also frequently used by Internet managers to verify correct operations of End Systems (ES) and to check that routers are correctly routing packets to the specified destination address.

    ICMP messages generated by router R1, in response to message sent by H0 to H1 and forwarded by R0. This message could, for instance be generated if the MTU of the link between R0 and R1 was smaller than size of the IP packet, and the packet had the Don't

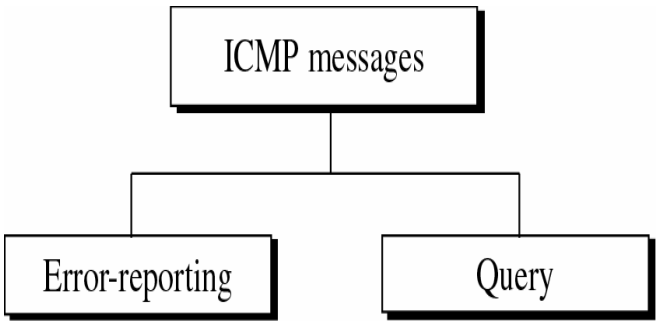
Fragment (DF) bit set in the IP packet header.The ICMP message is returned to H0, since this is the source address specified in the IP packet that suffered the problem.



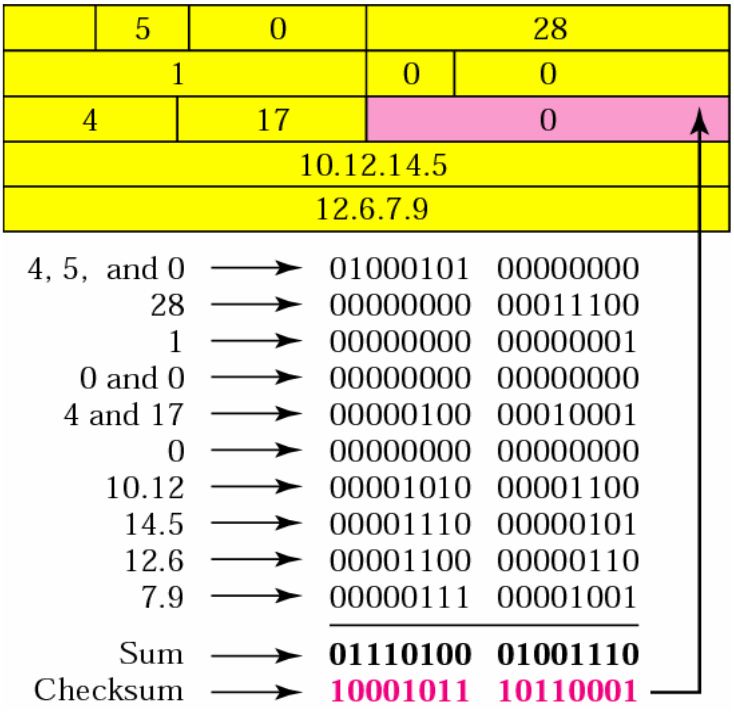
ICMP Messages

ICMP Messages are used by IP to send error and control messages ICMP uses IP to send messages. It does not report errors on ICMP messages. ICMP message are not required on datagram checksum errors. There are two types of messages namely

1) Error reporting
2) Query messages.

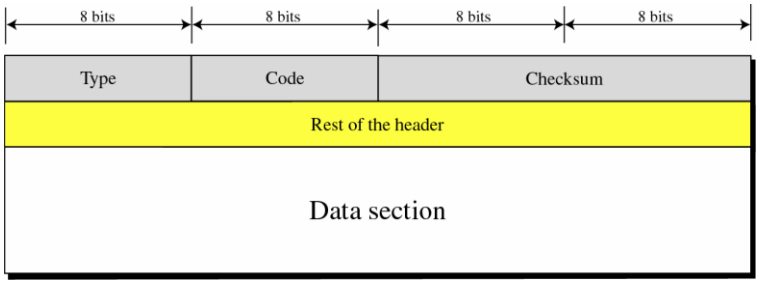


Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router solicitation or advertisement



General format of ICMP messages

An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all.



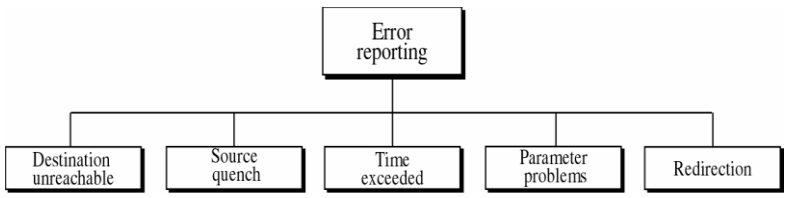
Type (8): specifies the type of ICMP message

Code (8): used to specify parameters of the message that can be encoded in a few bits

Checksum (16): hecksum of the entire ICMP message

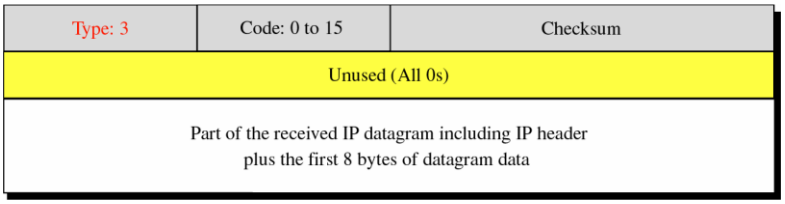
ERROR REPORTING

ICMP always reports error messages to the original source ICMP error messages report error conditions Typically sent when a datagram is discarded Error message is often passed from ICMP to the application program ICMP error essages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)



a) Destination-unreachable

Destination-unreachable messages with codes 2 or 3 can be created only by the destination host. Other destination-unreachable messages can be created only by routers. A router cannot detect all problems that prevent the delivery of a packet



Code 0 Net Unreachable

Code 1 Host Unreachable

Code 2 Protocol Unreachable

Code 3 Port Unreachable

Code 4 Fragmentation needed & Don't Fragment was set

Code 5 Source Route failed

Code 6 Destination Network Unknown

Code 7 Destination Host Unknown

Code 8 Source Host Isolated

Code 9 Communication Destination Network is Administratively Prohibited

Code 10 Communication Destination Host is Administratively Prohibited

Code 11 Destination Network Unreachable for Type of Service

Code 12 Destination Host Unreachable for Type of Service

Code 13 Communication Administratively Prohibited

Code Host Precedence Violation

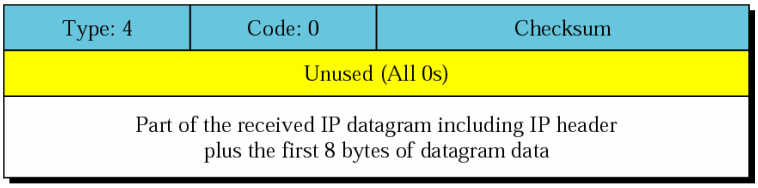
14

Code Precedence Cutoff Violation

15

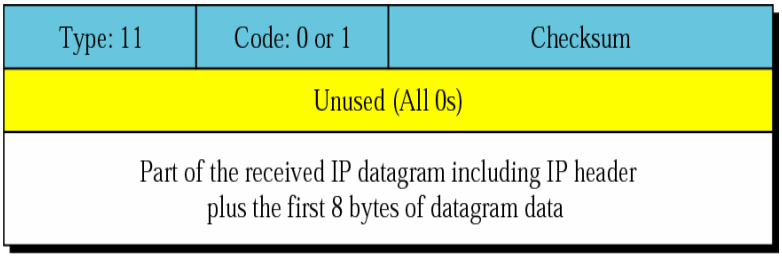
b) Source-quench

A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host. The source must slow down the sending of data grams until the congestion is relieved. One source-quench message should be sent for each datagram that is discarded due to congestion.



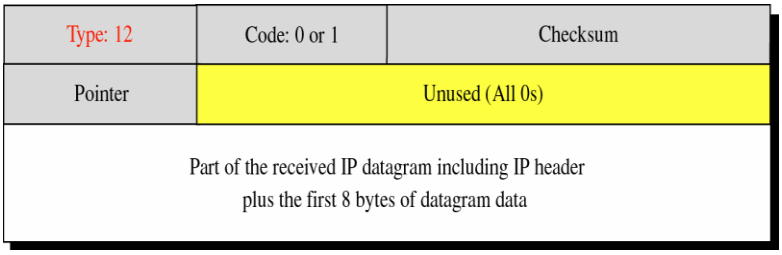
c) Time Exceed

Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source. When the final destination does not receive all of the fragments in a set time, it discards the received fragments and sends a time-exceeded message to the original source. In a time-exceeded message, code 0 is used only by routers to show that the value of the time-to-live field is zero. Code 1 is used only by the destination host to show that not all of the fragments have arrived within a set time.



d) Parameter Problem

A parameter-problem message can be created by a router or the destination host.

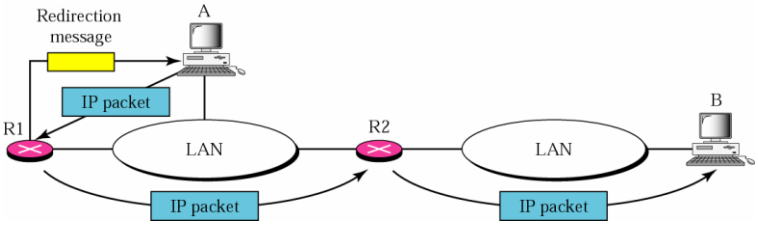


Code 0: Main header problem

Code 1: Problem in the option field

e) Redirection

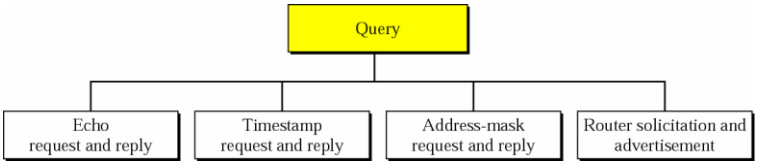
A host usually starts with a small routing table that is gradually augmented and updated. One of the tools to accomplish this is the redirection message. A redirection message is sent from a router to a host on the same local network.



Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

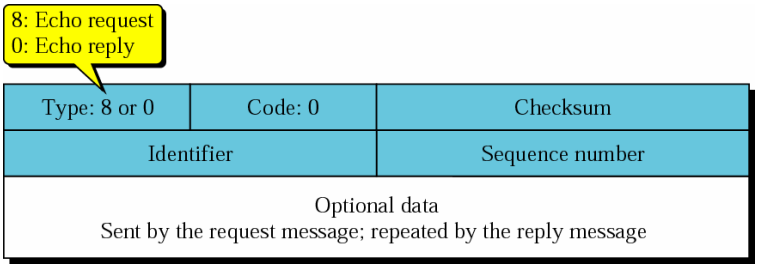
QUERY MESSAGES

ICMP can also diagnose some network problems through the query messages, a group of four different pairs of messages. In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.



a) Echo Request and Reply:

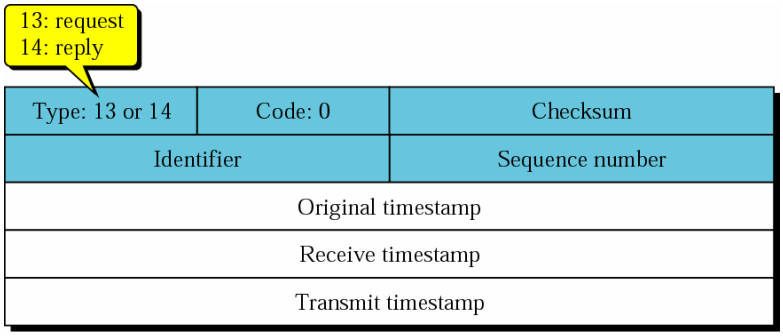
An echo-request message can be sent by a host or router. An echo-reply message is sent by the host or router which receives an echo-request message. Echo-request and echo-reply messages can be used by network managers to check the operation of the IP protocol. Echo-request and echo-reply messages can test the reach ability of a host. This is usually done by invoking the ping command.



b) Timestamp Request and Reply

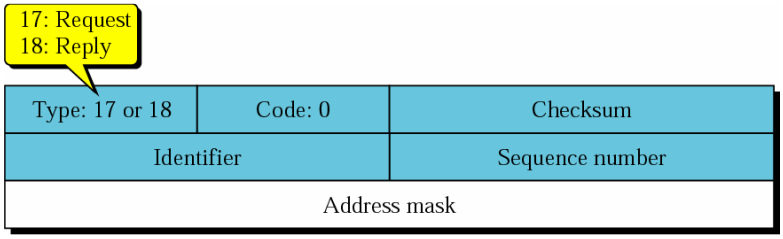
Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine even if their clocks are not

synchronized. The timestamp-request and timestamp-reply messages can be used to synchronize two clocks in two machines if the exact one-way time duration is known.



c) Address Mask Request & Address Mask Reply

A booting computer to determine the subnet mask in use on the local network uses the Address Mask Request ICMP Type 17. An intermediary device or computer acting as an intermediary device will reply with a Type 18 ICMP Address Mask Reply ICMP.

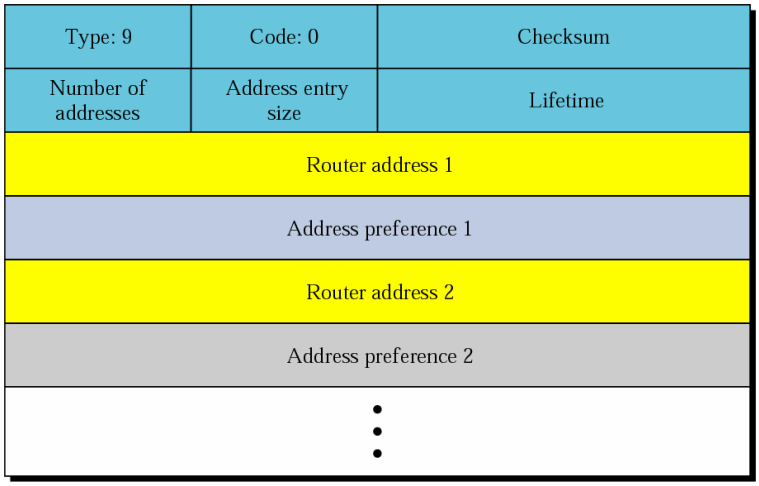


d) **Router-solicitation message**

Router discovery uses Internet Control Message Protocol (ICMP) router advertisements and router solicitation messages to allow a host to discover the addresses of operational routers on the subnet. Hosts must discover routers before they can send IP datagrams outside their subnet. Router discovery allows a host to discover the addresses of operational routers on the subnet. Each router periodically multicasts a router advertisement from each of its multicast interfaces, announcing the IP address of that interface. Hosts listen for advertisements to discover the addresses of their neighboring routers. When a host starts, it can send a multicast router solicitation to ask for immediate advertisements.

e) Router advertisement message

Router advertisement messages include a preference level and a lifetime field for each advertised router address. The preference level specifies the router's preference to become the default router. When a host chooses a default router address, it chooses the address with the highest preference. You can configure the preference level with the priority statement. The lifetime field indicates the maximum length of time that the advertised addresses are to be considered valid by hosts in the absence of further advertisements..



(OR)

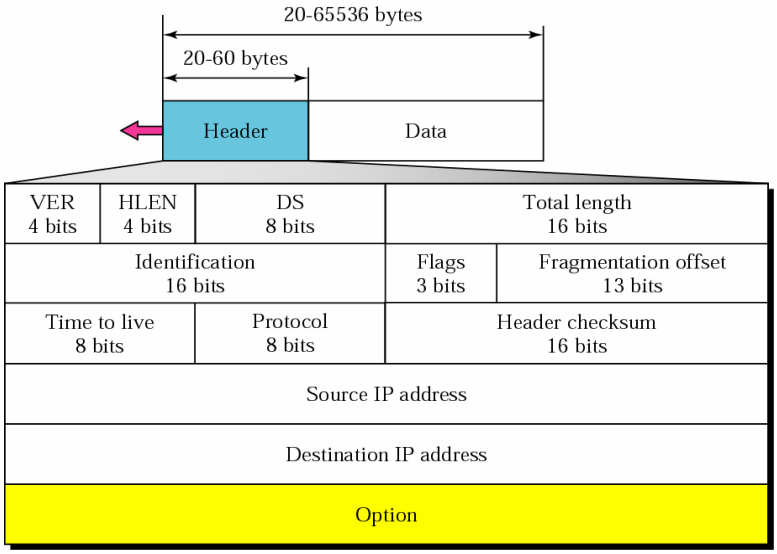
b) Discuss about IP Options in detail.

Ans:

The Internet Protocol (IP) is a network-layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is documented in RFC 791 and is the primary network-layer protocol in the Internet protocol suite. Along with the Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols. IP has two primary responsibilities: providing connectionless, best-effort delivery of datagram's through an internetwork; and providing fragmentation and reassembly of datagrams to support data links with different maximum-transmission unit (MTU) size

IP Datagram

A packet in the IP layer is called a datagram, a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery**.**



The following are the fields in IP Datagram

Version— Indicates the version of IP currently used.

***IP Header Length (*IHL)**—indicates the datagram header length in 32-bit words.

Type-of-Service—Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.

Identification—Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

***Flags*—**Consists of a 3-bit field of which the two low-order (least-significant) bits control **Fragmentation.** The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

Fragment Offset—Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

Time-to-Live—Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

***Protocol*—**Indicates which upper-layer protocol receives incoming packets after IP processing is complete.

***Header Checksum*—**helps ensure IP header integrity.

S*ource Address*—Specifies the sending node.

Destination Address—Specifies the receiving node..

***Options*—**Allows IP to support various options, such as security.

***Data*—**Contains upper-layer information.

Routing IP Datagram

If all computers were directly connected on the same physical network, there would be little need for the IP protocol. After all, so far in this description of the protocol, the only job IP has performed has been wrapping the transport layer packet into an IP datagram for transmission by the network level. In reality, an IP datagram sent between two computers on the public network typically passes through many different IP network devices along the way. It is the ability to route IP packets across different physical networks that is the heart of the Internet.

The public network, or Internet, is actually a collection of thousands of individual networks, interconnected together. These interconnections form a mesh network, creating millions of paths between the individual computers on the Internet. Routers are dedicated devices that are the interconnection point for the networks of the world. Routers are responsible for passing IP packets along from the source to the destination, across the various network interconnection points.

Each router that an IP packet passes through is referred to as a hop. In general, as the packet traverses the network, a router is only responsible for getting a packet to the next hop

along its path. Routers use the Internet and network layer. Routers need access to the network layer so they can physically receive packets. The network layer then passes the IP datagram up to the router IP layer. The router processes the destination address contained in the IP header and determines which device the send the IP packet on to, typically another router. The transport and user level data is not needed and is not unpacked from the IP datagram. This allows routers to function very quickly, as they are able to unpack the necessary information from the IP packet using specially designed hardware.

Routing Protocols

Routers are responsible for routing IP packets between a source and destination address. Typically, each router is responsible for only getting a packet to the next router along the path. As such, a router only needs to know the addresses of the routers to which it is directly connected. It also needs to know which connected router should be used for forwarding a packet. When the router examines the IP address of an incoming datagram, it accesses a database or table to determine which router should form the next hop in the path. Routers use various protocols to communicate with each other in order to set up the tables used to route packets.

Some common routing protocols include:

- Router Information Protocol (RIP)
- Open Shortest Path First (OSPF)
- Interior Gateway Routing Protocol (IGRP)
- Enhanced IGRP (EIGRP)
- Border Gateway Protocol (BGP)
- Intermediate System to Intermediate System (IS-IS)
- Constrained Shortest Path First (CSPF)

While routers operate on packets at the Internet layer, they also use transport layer services such as UDP and TCP to communicate with each other to build routing tables.

25. What is a Port? Explain the types of ports in detail.

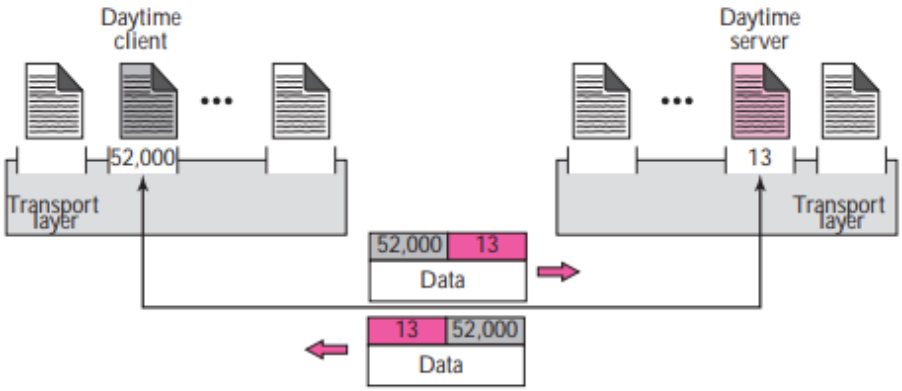
Ans: A remote computer can run several server programs at the same time, just as several local computers can run one or more client programs at the same time. For communication, we must define the

- ❑ Local host
- ❑ Local process
- ❑ Remote host
- ❑ Remote process

The local host and the remote host are defined using IP addresses. To define the processes, we need second identifiers called port numbers. In the TCP/IP protocol suite, the port

numbers are integers between 0 and 65,535. The client program defines itself with a port number, called the ephemeral port number. The word ephemeral means short lived and is used because the life of a client is normally short. An ephemeral port number is recommended to be greater than 1,023 for some client/server programs to work properly. The server process must also define itself with a port number. This port number, however, cannot be chosen randomly. If the computer at the server site runs a server process and assigns a random number as the port number, the process at the client site that wants to access that server and use its services will not know the port number. Of course, one solution would be to send a special packet and request the port number of a specific server, but this creates more overhead. TCP/IP has decided to use universal port numbers for servers; these are called well-known port numbers. There are some exceptions to this rule; for example, there are clients that are assigned well-known port numbers. Every client process knows the well-known port number of the corresponding server process. For example, while the daytime client process, discussed above, can use an ephemeral (temporary) port number 52,000 to identify itself, the daytime server process must use the well-known (permanent) port number 13.

Figure *Port numbers*



It should be clear by now that the IP addresses and port numbers play different roles in selecting the final destination of data. The destination IP address defines the host among the different hosts in the world. After the host has been selected, the port number defines one of the processes on this particular host.

ICANN Ranges

ICANN has divided the port numbers into three ranges: well-known, registered, and dynamic (or private).

❑ Well-known ports. The ports ranging from 0 to 1,023 are assigned and controlled by ICANN. These are the well-known ports.

❑ Registered ports. The ports ranging from 1,024 to 49,151 are not assigned or controlled by ICANN. They can only be registered with ICANN to prevent duplication.

❑ Dynamic ports. The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used as temporary or private port numbers. The original recommendation was that the ephemeral port numbers for clients be chosen from this range. However, most systems do not follow this recommendation.

(OR)

b) Write a detail note on IP Fragmentation.

Ans: If a router receives an IP packet that is too large for the network onto which the packet is being forwarded IP will fragment the original packet into smaller packets that will fit on the downstream network. When the packets arrive at their final destination, IP at the destination host reassembles the fragments into the original payload.

This process is referred to as fragmentation and reassembly. Fragmentation can occur in environments that have a mix of networking technologies, such as Ethernet and Token Ring. The fragmentation and reassembly works as follows:

1) When an IP packet is sent by the source, it places a unique value in the Identification field.
2) The IP packet is received at the router. The IP router notes that the maximum transmission unit (MTU) of the network onto which the packet is to be forwarded is smaller than the size of the IP packet.
3) IP fragments the original IP payload into fragments that will fit on the next network. Each fragment is Sent with its own IP header which contains

IP Header Field	Function
Source IP Address	The IP address of the original source of the IP datagram.
Destination IP Address	The IP address of the final destination of the IP datagram.
Identification	Used to identify a specific IP datagram and to identify all fragments of a specific IP datagram if fragmentation occurs.
Protocol	Informs IP at the destination host whether to pass the packet up to TCP, UDP, ICMP, or other protocols.
Checksum	A simple mathematical computation used to verify the integrity of the IP header.
Time to Live (TTL)	Designates the number of networks on which the datagram is allowed to travel before being discarded by a router. The TTL is set by the sending host and is used to prevent packets from endlessly circulating on an IP internetwork. When forwarding an IP packet, routers are required to decrease the TTL by at least one.

The original Identification field identifies all fragments that belong together. The More Fragments Flag indicates that other fragments follow. The More Fragments Flag is not set on the last fragment, because no other fragments follow it.

The Fragment Offset field indicates the position of the fragment relative to the original IP payload. When the fragments are received by IP at the remote host, they are identified by the Identification field as belonging together. The Fragment Offset is then used to reassemble the fragments into the original IP payload.

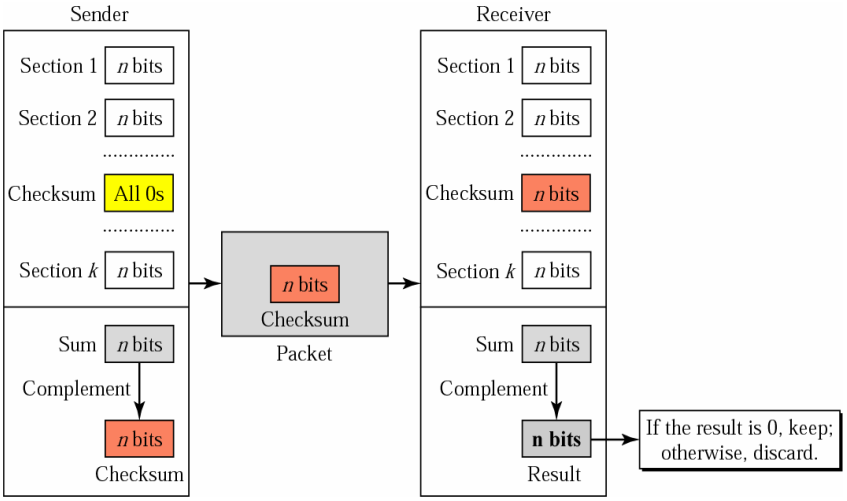
Check-Sum

The error detection method used by most TCP/IP protocols is called the checksum. The checksum protects against the corruption that may occur during the transmission of a packet. It is redundant information added to the packet.

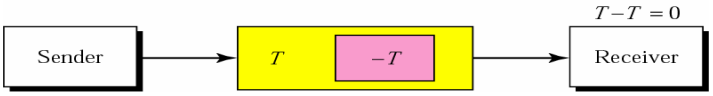
To create the checksum the sender does the following:

1) The packet is divided into k sections, each of n bits.
2) All sections are added together using 1's complement arithmetic.
3) The final result is complemented to make the checksum.

Checksum Concept:



Checksum in one's complement arithmetic



Fragmentation. The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

Fragment Offset—Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

Time-to-Live—Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

26. a) Discuss the need of IGMP in detail with the protocol format.

Ans: Internet Group management protocol (IGMP), a multicasting protocol in the internet protocols family, is used by IP hosts to report their host group memberships to any immediately neighboring multicast routers. IGMP messages are encapsulated in IP datagrams, with an IP protocol number of 2. IGMP has versions IGMP v1, v2 and v3

IGMPv1: Hosts can join multicast groups. There were no leave messages. Routers were using a time-out based mechanism to discover the groups that are of no interest to the members.

IGMPv2: Leave messages were added to the protocol. Allow group membership termination to be quickly reported to the routing protocol, which is important for high-bandwidth multicast groups and/or subnets with highly volatile group membership.

IGMPv3: Major revision of the protocol. It allows hosts to specify the list of hosts from which they want to receive traffic from. Traffic from other hosts is blocked inside the network. It also allows hosts to block inside the network packets that come from sources that sent unwanted traffic.

The variant protocols of IGMP are:

DVMRP: Distance Vector Multicast Routing Protocol.

IGAP: IGMP for user Authentication Protocol.

RGMP: Router-port Group Management Protocol.

Protocol Structure -IGMP (Internet Group Management Protocol)

There are basically 5 types of messages in the IGMP that must be implemented in IGMP for the IGMP v3 functional properly and be compatible with previous versions:

0x11: membership query

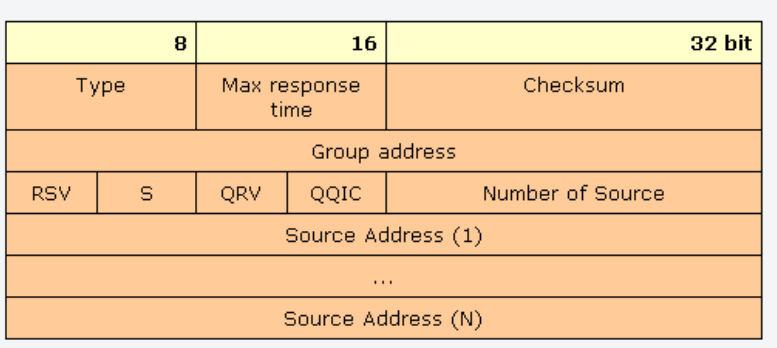
0x22: version 3 membership report

0x12: version 1 membership report

0x16: version 2 membership report

0x17 version 2 leave group

As an example, the message format for 0x11 (membership query) is displayed as follows



Type -- The message type: 0x11 (Membership query).

Max Response Time -- Used only in Membership query messages. Specifies the maximum time allowed before sending a responding report in units of 1/10 second. In all other messages, it is set to 0 by the sender and ignored by the receiver.

Checksum -- The checksum for messages errors

Group Address -- The Group address is set to 0 when sending a general query. It is set to the group address being queried, when sending a group specific query or group-and-source-specific query. In a membership report of a leave group message, it holds the IP multicast group address of the group being reported or left.

Group Record -- A block of fields containing information about the sender's membership in a single multicast group, on the interface from which the report is sent.

IGMP Messages

There are three message types used in IGMP. The IGMP 'type' field is set to the following values for each message type [in hex]:

MEMBERSHIP QUERY [0x11]

Membership Query messages are used by multicast enabled routers running IGMP to discover which hosts on attached networks are members of which multicast groups. Membership Query messages are sent to the 'all-systems' multicast group address of 224.0.0.1.

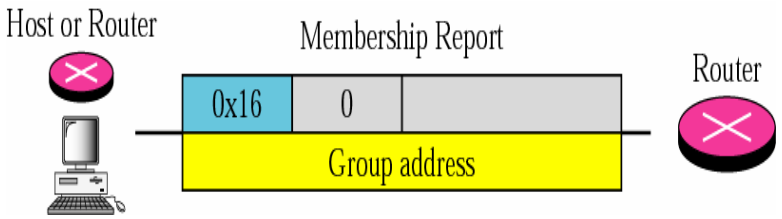
There are two types of Membership Queries:

General Query - used to learn which groups have members on an attached network.

Group-Specific Query - used to learn if a specific group has any members on an attached network.

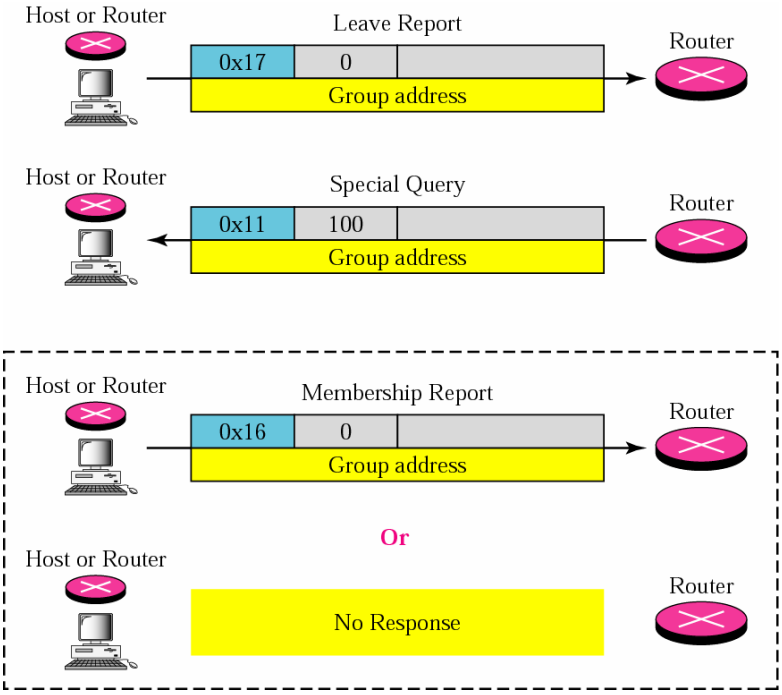
MEMBERSHIP REPORT (v1/v2) [0x12 / 0x16]

A membership report message is sent by a host whenever it joins a multicast group, and when responding to Membership Queries sent by an IGMP router that is functioning as a Querier.



LEAVE GROUP [0x17]

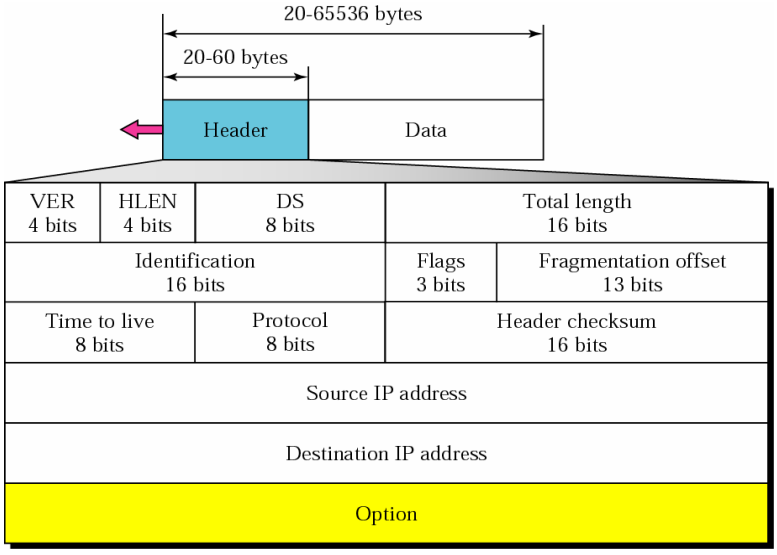
This message is sent when a host leaves a multicast group. This message is sent to the 'all-routers' multicast address of 224.0.0.2. The router then sends out a group-specific membership query to the network to verify if the last member of a group has left.



(OR)

b) Write in detail about IP Datagram Header.

Ans: A packet in the IP layer is called a datagram, a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery**.**



The following are the fields in IP Datagram

Version— Indicates the version of IP currently used.

***IP Header Length (*IHL)**—indicates the datagram header length in 32-bit words.

Type-of-Service—Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.

Identification—Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

***Flags*—**Consists of a 3-bit field of which the two low-order (least-significant) bits control **Fragmentation.** The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

Fragment Offset—Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

Time-to-Live—Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

***Protocol*—**Indicates which upper-layer protocol receives incoming packets after IP processing is complete.

***Header Checksum*—**helps ensure IP header integrity.

S*ource Address*—Specifies the sending node.

Destination Address—Specifies the receiving node..

***Options*—**Allows IP to support various options, such as security.

***Data*—**Contains upper-layer information.

Routing IP Datagram

If all computers were directly connected on the same physical network, there would be little need for the IP protocol. After all, so far in this description of the protocol, the only job IP has performed has been wrapping the transport layer packet into an IP datagram for transmission by the network level. In reality, an IP datagram sent between two computers on the public network typically passes through many different IP network devices along the way. It is the ability to route IP packets across different physical networks that is the heart of the Internet.

The public network, or Internet, is actually a collection of thousands of individual networks, interconnected together. These interconnections form a mesh network, creating millions of paths between the individual computers on the Internet. Routers are dedicated devices that are the interconnection point for the networks of the world. Routers are responsible for passing IP packets along from the source to the destination, across the various network interconnection points.

Each router that an IP packet passes through is referred to as a hop. In general, as the packet traverses the network, a router is only responsible for getting a packet to the next hop along its path. Routers use the Internet and network layer. Routers need access to the network layer so they can physically receive packets. The network layer then passes the IP datagram up to the router IP layer. The router processes the destination address contained in the IP header and determines which device the send the IP packet on to, typically another router. The transport and

user level data is not needed and is not unpacked from the IP datagram. This allows routers to function very quickly, as they are able to unpack the necessary information from the IP packet using specially designed hardware.

Routing Protocols

Routers are responsible for routing IP packets between a source and destination address. Typically, each router is responsible for only getting a packet to the next router along the path. As such, a router only needs to know the addresses of the routers to which it is directly connected. It also needs to know which connected router should be used for forwarding a packet. When the router examines the IP address of an incoming datagram, it accesses a database or table to determine which router should form the next hop in the path. Routers use various protocols to communicate with each other in order to set up the tables used to route packets.

Some common routing protocols include:

- Router Information Protocol (RIP)
- Open Shortest Path First (OSPF)
- Interior Gateway Routing Protocol (IGRP)
- Enhanced IGRP (EIGRP)
- Border Gateway Protocol (BGP)
- Intermediate System to Intermediate System (IS-IS)
- Constrained Shortest Path First (CSPF)

While routers operate on packets at the Internet layer, they also use transport layer services such as UDP and TCP to communicate with each other to build routing tables.

Reg.No ----------

[17CTU403]

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)

(Established Under Section 3 of UGC Act 1956)

COIMBATORE – 641 021

B.Sc. DEGREE EXAMINATION

Fourth Semester - III Internal Examination

March 2019

Computer Technology

Internetworking with TCP/IP

Time: 2 Hrs **Max.Marks: 50**

Date : **Class: II B.Sc (CT)**

Part - A **(20 X 1 = 20)**

Answer All the Questions

1. Which of the following protocol is used for dynamic addressing?

   a) BOOTP b) DHCP c) RARP d) ARP

2. The Process to Process Communication is done in ____ layer

   a) Network b) datalink c) transport d) Application

3. _____ is a protocol used with Connection less service in Transport Layer.

   a) ICMP b) DHCP c) UDP d) TCP

4. When an IP address is given which of the following returns physical Address?

   a) IGMP b) ICMP c) RARQ d) ARQ

5. TCP supports ___type of Service.

   a) Connection Oriented b) Connection less c) both d) Others

6. DHCP Stands for _____

7. BOOTP stands for _____

8. UDP stands for _____

9. TCP stands for _____

10. SMTP stands for _____

11. FTP uses the service of _____

    a) TCP b) UDP c) Both d) None

12. IP address cannot be represented in _____format.

    a) Hexadecimal b) Dotted Decimal c) Octal d) Binary

13. An IPv4 address is represented in ___ bits

    a) 32 b) 64 c) 128 d) 16

14. VPN stands for Virtual Private Network

15. _____ refers to the number of systems that are not used in a network.

    a) Hide b) Mask c) Multi d) No-operation

16. NAT represents Network Address Translation

17. POP stands for Post Office Protocol

18. Which of the following is used for naming service of a server?

    a) DNS b) DBS c) DHCP d) BOOTP

19. The methods of delivering message to all the users is called ___

    a) Unicasting b) Multicasting c) broadcasting d) None

20. The methods of delivering message to more number of users is called and not to all is called ___

    a) Unicasting b) Multicasting c) broadcasting d) None

Part - B **(3 X 2 = 6)**

Answer all the Questions

21. List the functions of Transport Layer.

22. Differentiate BOOTP & DHCP.

23. What is SMTP?

Part - C **(3 X 8 = 24)**

Answer all the Questions

24. a) Describe the role of DHCP in Networking.

(OR)

b) Discuss the role of Transport layer in detail along with UDP.

25. a) Write about DNS in detail.

(OR)

b) Write in general about the various mail scenarios.

26. a) What is a Telnet? Write a detail note on it.

(OR)

b) Discuss in detail about VPN.

Reg.No ----------

[17CTU403]

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)

(Established Under Section 3 of UGC Act 1956)

COIMBATORE – 641 021

B.Sc. DEGREE EXAMINATION

Fourth Semester - III Internal Examination

March 2019

Computer Technology

Internetworking with TCP/IP

Time: 2 Hrs **Max.Marks: 50**

Date : **Class: II B.Sc (CT)**

Part - A **(20 X 1 = 20)**

Answer All the Questions

1. Which of the following protocol is used for dynamic addressing?

   a) BOOTP b) **DHCP** c) RARP d) ARP

2. The Process to Process Communication is done in ____ layer

   a) Network b) datalink c) **transport** d) Application

3. _____ is a protocol used with Connection less service in Transport Layer.

   a) ICMP b) DHCP c) **UDP** d) TCP

4. When an IP address is given which of the following returns physical Address?

   a) IGMP b) ICMP c) **RARQ** d) ARQ

5. TCP supports ___type of Service.

   a) **Connection Oriented** b) Connection less c) both d) Others

6. DHCP Stands for Dynamic Host Configuration Protocol

7. BOOTP stands for Bootstrap Protocol

8. UDP stands for User Datagram Protocol

9. TCP stands for Transmission Control Protocol

10. SMTP stands for Simple Mail Transfer Protocol

11. FTP uses the service of _____

    a) **TCP** b) UDP c) Both d) None

12. IP address cannot be represented in _____format.

    a) Hexadecimal b) Dotted Decimal c) **Octal** d) Binary

13. An IPv4 address is represented in ___ bits

    a) **32** b) 64 c) 128 d) 16

14. VPN stands for Virtual Private Network

15. _____ refers to the number of systems that are not used in a network.

    a) Hide b) **Mask** c) Multi d) No-operation

16. NAT represents Network Address Translation

17. POP stands for Post Office Protocol

18. Which of the following is used for naming service of a server?

    a) **DNS** b) DBS c) DHCP d) BOOTP

19. The methods of delivering message to all the users is called ___

    a) Unicasting b) Multicasting c) **broadcasting** d) None

20. The methods of delivering message to more number of users is called and not to all is called ___

    a) Unicasting b) **Multicasting** c) broadcasting d) None

Part - B **(3 X 2 = 6)**

Answer all the Questions

21. List the functions of Transport Layer.

    Ans: Process to Process Communication, Flow Control, Error Control and Congestion Control.

22. Differentiate BOOTP & DHCP.

    Ans: Bootp is a static configuration Protocol while DHCP is a Dynamic configuration protocol.

23. What is SMTP?

    Ans: It's a mail transfer protocol used between users of same and different networks.

Part - C **(3 X 8 = 24)**

Answer all the Questions

24. a) Describe the role of DHCP in Networking.

    Ans: The **Dynamic Host Configuration Protocol (DHCP)** has been devised to provide static and dynamic address allocation that can be manual or automatic.

    Static Address Allocation

    In this capacity DHCP acts like BOOTP. It is backward compatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server. A DHCP server has a database that statically binds physical addresses to IP addresses.

    Dynamic Address Allocation

    DHCP has a second database with pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available IP addresses and assigns an IP address for a negotiable period of time.

When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database. The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (like a subscriber to a service provider). DHCP provides temporary IP address for a limited period of time.

The addresses assigned from the pool are temporary addresses. The DHCP server issues a lease for a specific period of time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the choice to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

Manual and Automatic Configuration

One major problem with BOOTP protocol is that the table mapping the IP addresses to physical addresses needs to be manually configured. This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes. DHCP, on the other hand, allows both manual and automatic configurations. Static addresses are created manually; dynamic addresses are created automatically.

Packet Format

To make DHCP backward compatible with BOOTP, the designers of DHCP have decided to use almost the same packet format. They have only added a 1-bit flag to the packet. However, to allow different interactions with the server, extra options have been added to the option field. Figure 16.6 shows the format of a DHCP message. The new fields are as follows:

Flag. A 1-bit flag has been added to the packet (the first bit of the unused field) to let the client specify a forced broadcast reply (instead of unicast) from the server. If the reply were to be unicast to the client, the destination IP address of the IP packet is the address assigned to the client. Since the client does not know its IP address, it may discard the packet. However, if the IP datagram is broadcast, every host will receive and process the broadcast message.

Options. Several options have been added to the list of options. One option, with value 53 for the tag subfield (see figure 16.5), is used to define the type of interaction

between the client and server.(see table 16.2). Other options define parameters such as lease time and so on. The options field in DHCP can be up to 312 bytes.

Figure 16.6 ***DHCP packet***

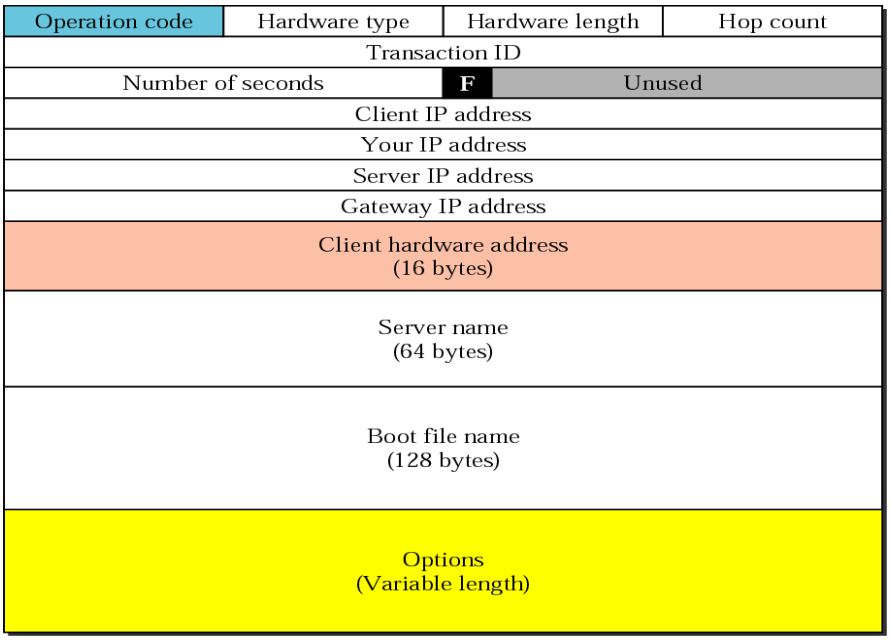


Table 16.2 *Options for DHCP*

Value	*Value*
1 DHCPDISCOVER	5 DHCPACK
2 DHCPOFFER	6 DHCPNACK
3 DHCPREQUEST	7 DHCPRELEASE
4 DHCPDECLINE	

We will see the use of these options in the next section.

Transition states

The DHCP client transitions from one state to another depending on the messages it receives or sends. See figure 16.7.

Initializing state

When the DHCP client first starts, it is in the initializing state. The client broadcasts a DHCPDISCOVER message (a request message with the DHCPDISCOVER option) using port 67.

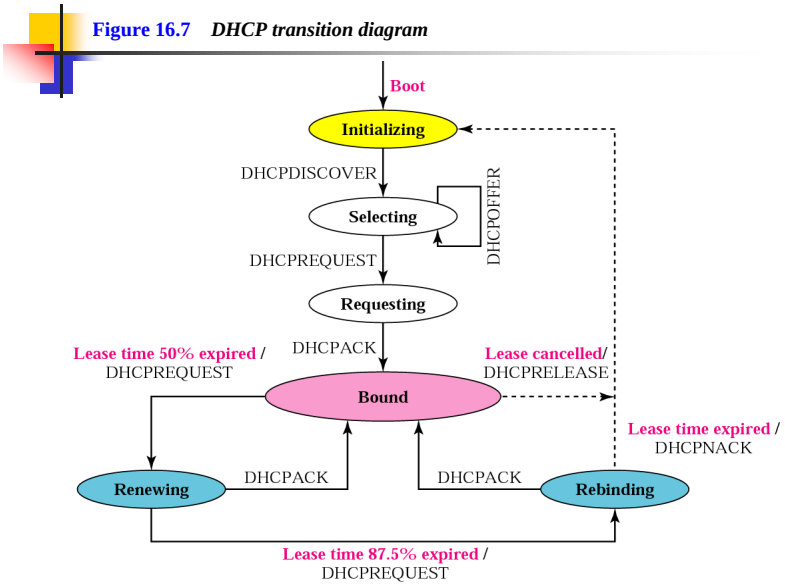


Figure 16.7 *DHCP transition diagram*

Selecting state

After sending the DHCPDISCOVER message, the client goes to the selecting state. Those servers that can provide this type of service respond with a DHCPOFFER message. In these messages, the servers offer an IP address. The can also offer the lease duration. The default is 1 h. The server that sends a DHCPOFFER locks the offered IP address so that it is not available to any other clients. The client chooses one of the offers and sends a DHCPREQUEST message to be selected server. It then goes to the requesting state. However, if the client receives no DHCPOFFER message, it tries four more times, each with a span of 2s. If there is no reply to any of these DHCPDISCOVER, the client sleeps for 5 minutes before trying again.

Requesting state

The client remains in the requesting state until it receives a DHCPPACK message from the server which creates the binding between the client physical address and its IP address. After the receipt of the DHCPACK, the client goes to the bound state

Selectijtate

The client remains in the renewing state until one of two events happens. It can receive a DHCPACK, which renews the least agreement. In this case, the client resets and goes back to the bound state. Or, if a DHCPACK is not received, and 87.5% of the lease time expires; the client goes to the rebinding state.

Rebinding state

The client remains in the rebinding state until one of three events happens. If the client receives a CPNACK or the lease expires, it goes back to the initializing state and tries to get another IP address. If the client receives a DHCPACK it goes to the bound state and resets the timer.

Exchanging Messages

Figure 16.8 *Exchanging messages*

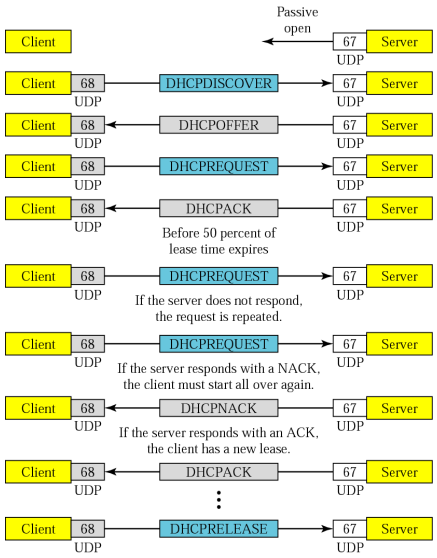
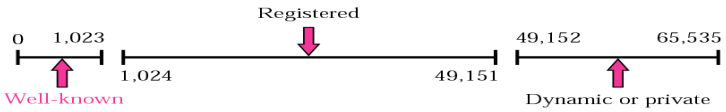


Figure 11.5 *ICANN ranges*



- **Well-known ports**. The ports ranging from 0 to 1,023 are assigned and controlled by ICANN. These are the well-known ports.
- **Registered ports.** The ports ranging from 1,024 to 1,023 are not assigned or controlled by ICANN. They can only be registered with ICANN to prevent duplication.

  Dynamic ports. The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used as temporary or private port numbers. The original recommendation was that the ephemeral port numbers for clients to be chosen from this range. However, most systems do not follow this recommendation.

(OR)

b) Discuss the role of Transport layer in detail along with UDP.

Ans: UDP uses concept common to the transport layer. These concepts will be discussed here briefly, and then expanded in the next chapter on the TCP protocol.

Connectionless Services

As mentioned previously, UDP provides a **connectionless service.** This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagram's even if they are coming from the

same source process and going to the same destination program. The user datagram's are not numbered. Also, there is no connection establishment and no connection termination as is the case for TCP. This means that each user datagram can travel on a different path.

One of the ramifications of being connectionless is that the process that uses UDP cannot send a stream of data to UDP and expect UDP to chop them into different related user data grams. Instead each request must be small enough to fit into one user datagram. Only those processes sending short messages should use UDP.

Flow and Error Control

UDP is a very simple, unreliable transport protocol. There is no flow control, and hence no window mechanism. The receiver may overflow with incoming messages. There is no error control mechanism in UDP except for the checksum. This means thus the sender does not know if message has been lost are duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded.

The lack of **flow control and error control** means that the process using UDP should provide for these mechanisms.

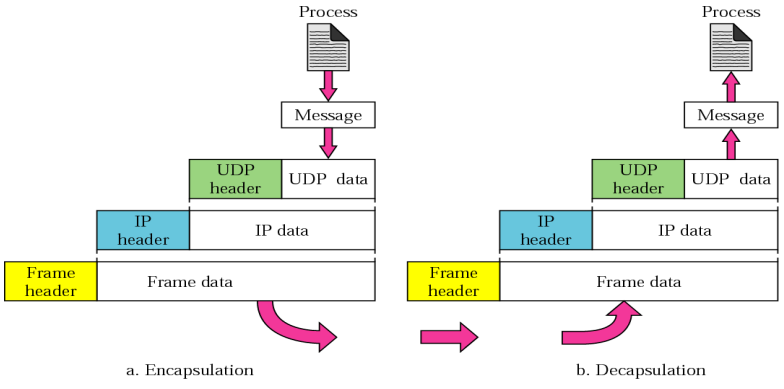
Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages (see figure 11.10).

Encapsulation

When a process has a message to send through UDP, it passes the message to UDP along with a pair of socket addresses and the length of data. UDP receives the data and adds the UDP header.

Figure 11.10 *Encapsulation and decapsulation*



UDP then passes the user datagram to IP with the socket addresses. IP adds its own header, using the value 17 in the protocol field,indicating that the data has come from the UDP protocol. The IP datagram is then passed to the data link layer. The data link layer receives the IP datagram, adds its own header(and possibly a trailer), and passes it to the physical layer. The physical layer encodes the bits into electrical or optical signals and sends it to the remote machine.

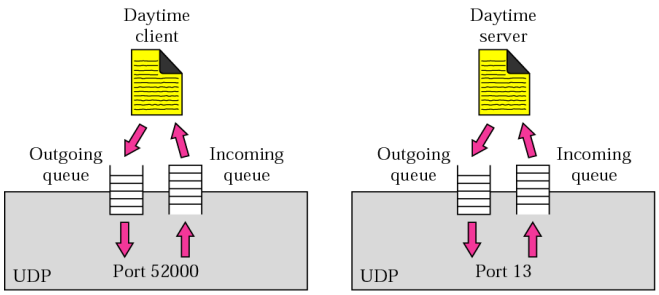
Decapsulation :

When the message arrives at the destination host, the physical layer decodes the signals into bits and passes it to the data link layer. The data link layer uses the header (and the trailer) to check the data. If there is no error, the header and trailer are dropped and the datagram is passed to **IP**. The IP software does its own checking. If there is no error, the header is dropped and the user datagram is passed to UDP with the sender and receiver IP addresses. UDP uses the checksum to check the entire user datagram. If there is no error, the header is dropped and the application data along with sender socket address is passed to the process. The sender socket address is passed to the process in case it needs to respond to the message received.

Queuing

We have talked about ports without discussing the actual implementation of them. In UDP queues are associated with ports (see figure 11.11). At the client site, when a process starts, it requests aport number from the operating systems. Some implementations create both an incoming and an outgoing queue associated with each process. Other implementations create only an incoming queue associated with each process.

Figure 11.11 ***Queues in UDP***



Note that even if a process wants to communicate with multiple processes, it obtains only one port number and eventually one outgoing and one incoming queue. The queues opened by the client are, in most cases, identified by ephemeral port numbers.

The queues function as long as the process is running. When the process terminates, the queues are destroyed.

The client process can send messages to the outgoing queue by using the source port number specified in the request. UDP removes the message one by one, and, after adding the UDP header, delivers them to IP. An outgoing queue can overflow. If this happens, the operating systems can ask the client process to wait before sending any more messages.

When a message arrives for a client, UDP checks to see if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue.

If there is no such queue, UDP discards the user datagram and asks the ICMP protocol to send a port unreachable message to be sent to the server. All of the incoming messages for one particular client program, whether coming from the same or different server, are sent to the same queue. An incoming queue can overflow. If this happens, UDP drops the user datagram and asks for a port unreachable message to be sent to the server.

At the server site, the mechanisms of creating queues are different. In its simplest form, a server asks for incoming and outgoing queues using its well-known port when it starts running. The queues remain open as long as the server is running.

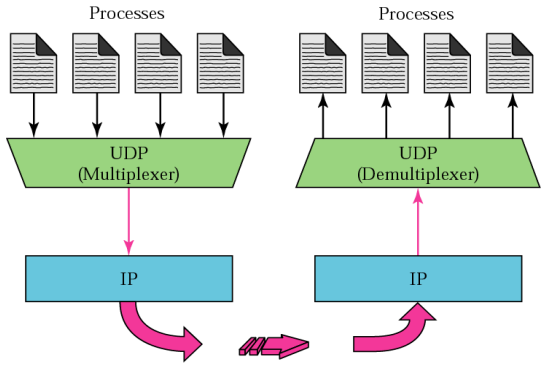
When a message arrives for a server, UDP checks to see if an incoming queue has been created for the port number specified in destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue. If there is no such a queue, UDP discards the user datagram and asks the ICMP protocol to send a port unreachable message to the client. All of the incoming messages for one particular server, whether coming from the same or a different client, are sent to the same queue. An incoming queue can overflow. If this happens, UDP drops the user datagram and asks for a port unreachable message to be sent to the client.

When a server wants to respond to a client, it sends messages to the outgoing queue using the source port number specified in the request. UDP removes the message one by one, and , after adding the UDP header, delivers them to IP. An outgoing queue can overflow. If this happens, the operating system asks the server to wait before sending any more messages.

Multiplexing and Demultiplexing

In a host running a TCP/IP protocol suite, there is only one UDP but possibly several processes that may want use the services of UDP. To handle this situation, UDP multiplexes and demultiplexes(see figure 11.12).

Figure 11.12 ***Multiplexing and demultiplexing***



Multiplexing

At the sender site, there may be several processes that need to send user datagram's. However, there is only one UDP. This is a many-to-one relationship and requires multiplexing. UDP accepts messages from different processes, differentiated by their assigned port numbers. After adding the header, UDP passes the user datagram to the IP.

Demultiplexing

At the receiver site, there is only one UDP. However, we may have many processes that can receive user datagram's. This is a one- to-many relationship and requires demultiplexing. UDP receives user datagrams from IP. After error checking and dropping of the header, UDP delivers each message to the appropriate process based on the port numbers.

25. a) Write about DNS in detail.

Ans: To identify an entity, TCP\IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name.

When the Internet was small, mapping was done using a *host file.* The host file had only two columns: name and address. Every host could store the host file on its disk and update it periodically from a master host file. When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.

Today, however, it is impossible to have one single host file to relate every address with a name and vice versa. The host file would be too large to store in every host. In addition, it would be impossible to update all the host files every time there is a change.

One solution would be to store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping, But we know that this would create a huge amount of traffic on the Internet.

Another solution , the one used today, is to divide this huge amount of information into smaller parts and store each part on a different computer. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS)

NAME SPACE

To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses. In other words, the names must be unique because the addresses are unique. A name space that maps each addresses to a unique name can be organized in two ways: flat or hierarchical.

Flat Name Space

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section: if they do, it has no meaning. The main disadvantages of a flat name space are that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication. Hierarchical Name Space

In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, and the third part can define departments in the organization, and do on. In this case, the authority to assign and control the name spaces can be decentralized. A central authority can assign the part of the name that defines the nature of the organization and the name of the organization. The responsibility of the rest of the name can be given to the organization itself. The organization can add suffixes (or prefixes) to the name to define its host or recourses. The management of the organization need not worry that the prefix chosen for a host is taken by another organization because, even if part of an address is the same, the whole address is different. For example, assume two colleges and a company call one of their computers challenger. The first college is given a name by the central authority such as fhda.edu; the second college is given the name smart.com. When each of these organizations adds the name challenger to the name they have already been given, the end result is three distinguishable names: challenger.fhda.edu, challenger.berkely.edu, and challenger.smart.com. The names

are unique without the need for assignment by a central authority. The central authority controls only part of the name, not the whole.

DOMAIN NAME SPACE

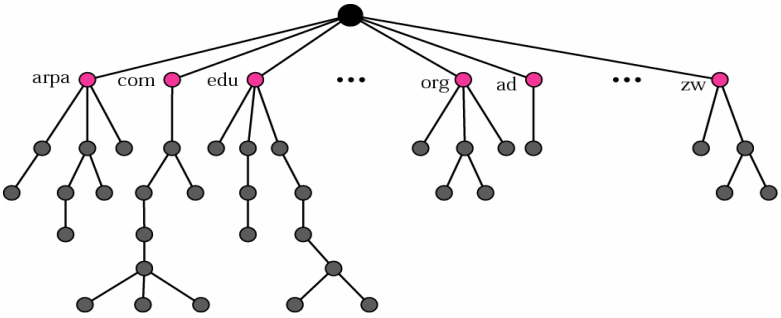
To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127 (see Figure).

Label

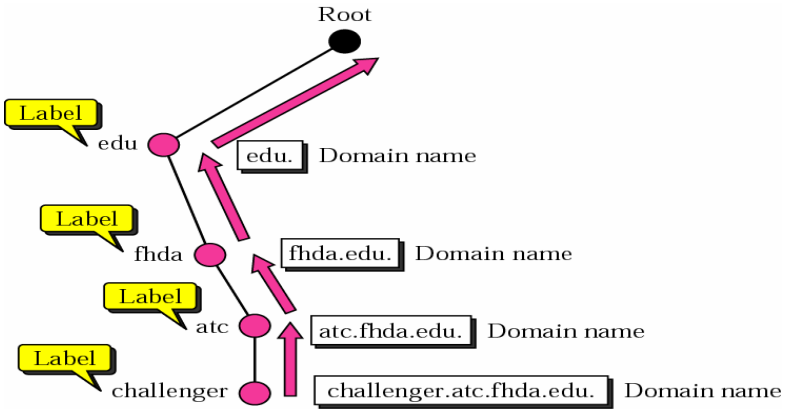
Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name

Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root.



The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.



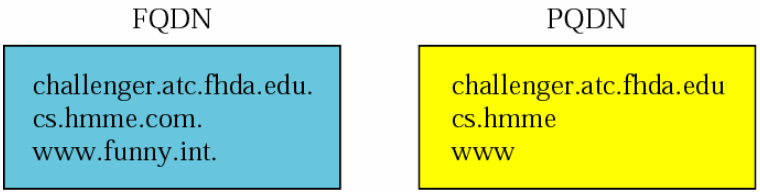
Fully Qualified Domain Name (FQDN)

If a label is terminated by a null string, it is called a fully qualified domain name (FQDN). An FQDN is a domain name that contains the full name of a host. It contains all labels, from the most specific to the most general, that uniquely define the name of the host.

For example, the domain name challenger.atc.fhda.edu. is the FQDN of a computer named challenger installed at the Advanced Technology Center (ATC) at De Anza College. A DNS server can only match an FQDN to an address. Note that the name must end with a null label, but because null means nothing, the label ends with a dot (.).

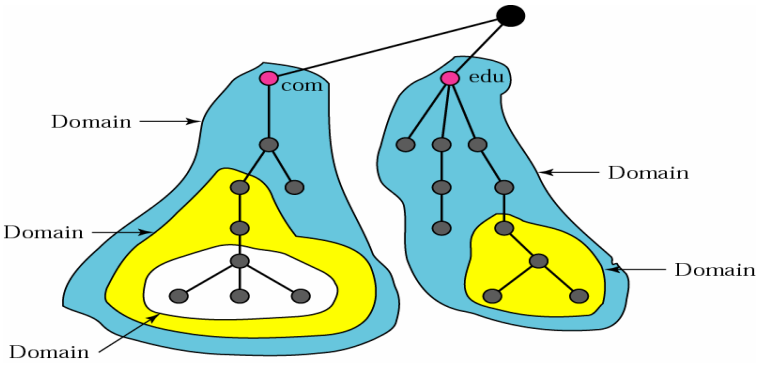
Partially Qualified Domain Name (PQDN)

If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN). A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client. Here the resolver can supply the missing part, called the suffix, to create an FQDN. For example, if a user at the fhda.edu. Site wants to get the IP address of the challenger computer, he or she can define the partial name



Domain

A domain is a sub-tree of the domain name space. The name of the domain is the domain name of the node at the top of the sub-tree. The above figure shows some domains. Note that a domain may itself be divided into domains (or subdomains as they are sometimes called).



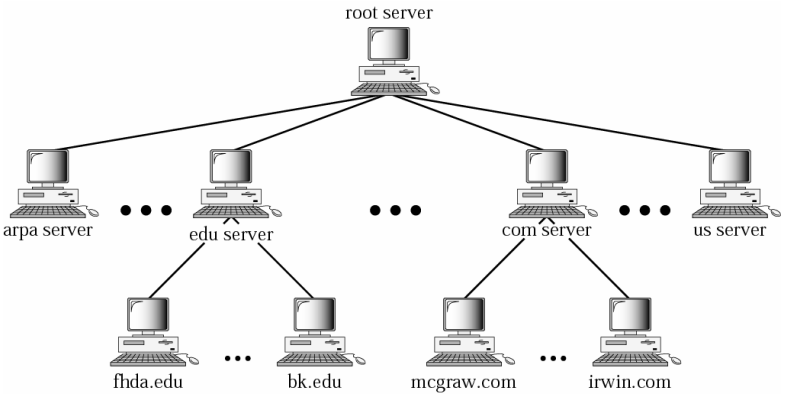
DISTRIBUTION OF NAME SPACE

The information contained in the domain space must be stored. However, it is very inefficient and also not reliable to have just one computer store such huge amount of information. It is inefficient because responding to request from all over the world places a heavy load on the system. It is not reliable because any failure makes the data inaccessible.

Hierarchy of Name Servers

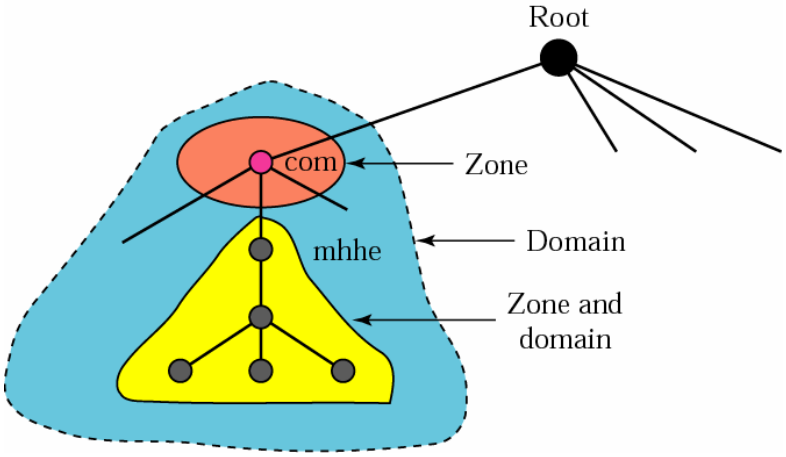
The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level. In other words, we let the root stand alone and create as many domains

(subtrees) as there are first-level nodes. Because a domain created this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or small domain. In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names (see below figure).



Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree. If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the "domain" and the "zone" refer to the same thing. The server makes a database called a zone file and keeps all the information for every node under that domain. However, if a server divides its domain into sub-domains and delegates part of its



Authority to others servers, "domain" and "zone" refer to different things. The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers. Of course the original server does not free itself from responsibility totally: It still has a zone, but the detailed information is kept by the lower-level servers.

A server can also divide part of its domain and delegate responsibility but still keep part of the domain for itself. In this case, its zone is made of detailed information for the part of the domain that is not delegated and references to those parts that are delegated.

Root Server

A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The servers are distributed all around the world.

Primary and Secondary Servers

DNS defines two types of servers: primary and secondary. A primary server is a server that stores a file about the zone for which it is an authority. It is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.

A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary.

The primary and secondary servers are both authoritative for the zones they serve. The idea is not to put the secondary server at a lower level of authority but to create redundancy for the data so that if one server fails, the other can continue serving clients. Note also that a server can be a primary server for a specific zone and a secondary server for another zone. Therefore, when we refer to a server as a primary or secondary server, we should be careful to which zone we refer.

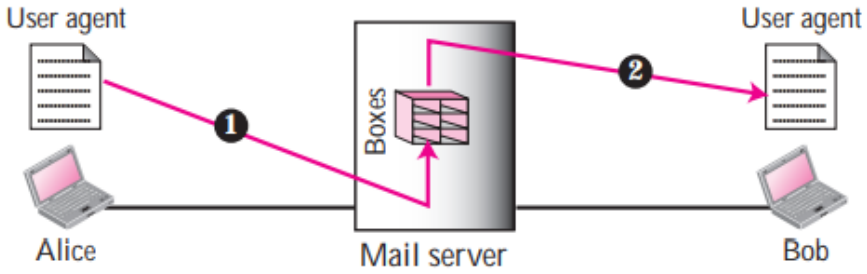
(OR)

b) Write in general about the various mail scenarios.

Ans:

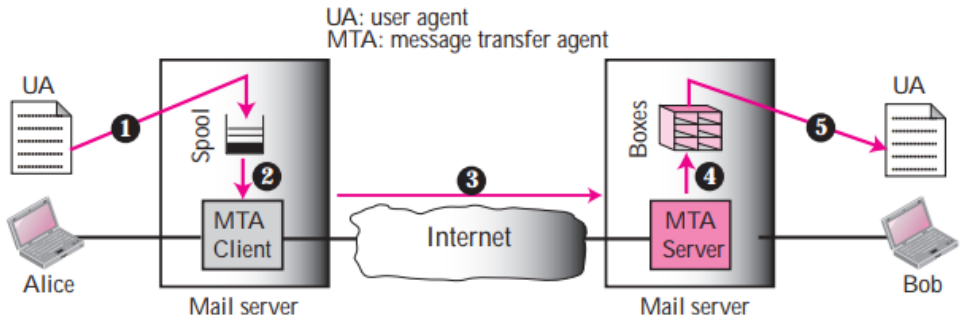
First Scenario: In the first scenario, the sender and the receiver of the e-mail are users (or application programs) on the same mail server; they are directly connected to a shared mail server. The administrator has created one mailbox for each user where the received messages are stored. A mailbox is part of a local hard drive, a special file with permission restrictions. Only the owner of the mailbox has access to it. When Alice needs to send a message to Bob, she runs a user agent (UA) program to prepare the message and store it in Bob's mailbox. The message has the sender and recipient mailbox addresses (names of files). Bob can retrieve and read the contents of his mailbox at his convenience using a user agent.

First scenario



Second Scenario: In the second scenario, the sender and the receiver of the e-mail are users (or application programs) on two different mail servers. The message needs to be sent over the Internet. Here we need user agents (UAs) and message transfer agents (MTAs). Alice needs to use a user agent program to send her message to the mail server at her own site. The mail server at her site uses a queue (spool) to store messages waiting to be sent. Bob also needs a user agent program to retrieve messages stored in the mailbox of the system at his site. The message, however, needs to be sent through the Internet from Alice's site to Bob's site. Here two message transfer agents are needed: one client and one server. Like most client-server programs on the Internet, the server needs to run all of the time because it does not know when a client will ask for a connection. The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent.

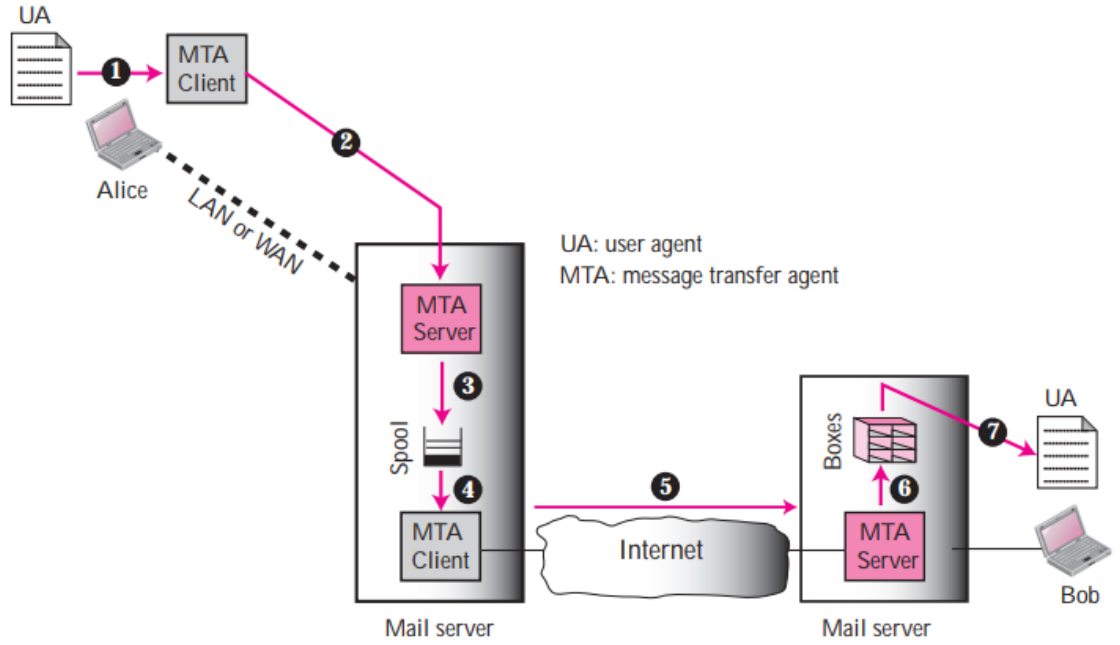
Second scenario



Third Scenario: Bob, as in the second scenario, is directly connected to his mail server. Alice, however, is separated from her mail server. Alice is either connected to the mail server via a point-to-point WAN—such as a dial-up modem, a DSL, or a cable modem—or she is connected to a LAN in an organization that uses one mail server for handling e-mails; all users need to send their messages to this mail server. Alice still needs a user agent to prepare her message. She then needs to send the message through the LAN or WAN. This can be done through a pair of message transfer agents (client and server). Whenever Alice has a

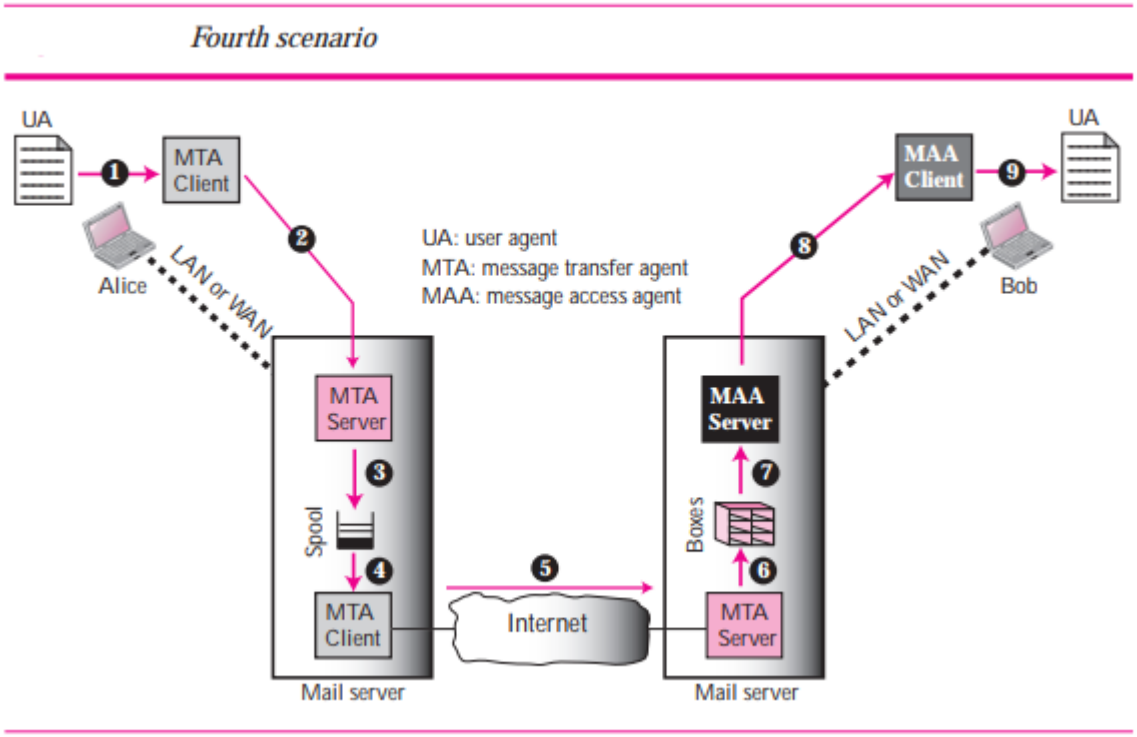
message to send, she calls the user agent which, in turn, calls the MTA client. The MTA client establishes a connection with the MTA server on the system, which is running all the time. The system at Alice's site queues all messages received. It then uses an MTA client to send the messages to the system at Bob's site; the system receives the message and stores it in Bob's mailbox.

Third scenario



Fourth Scenario:

In the fourth and most common scenario, Bob is also connected to his mail server by a WAN or a LAN. After the message has arrived at Bob's mail server, Bob needs to retrieve it. Here, we need another set of client-server agents, which we call message access agents (MAAs). Bob uses an MAA client to retrieve his messages. The client sends a request to the MAA server, which is running all the time, and requests the transfer of the messages. The situation is shown in Figure 23.4. There are two important points we need to emphasize here. First, Bob cannot bypass the mail server and use the MTA server directly. To use the MTA server directly, Bob would need to run the MTA server all the time because he does not know when a message will arrive. This implies that Bob must keep his computer on all the time if he is connected to his system through a LAN. If he is connected through a WAN, he must keep the connection up all the time. Neither of these situations is feasible today. Second, note that Bob needs another pair of client-server programs: message access programs. This is because an MTA client-server program is a push program: the client pushes the message to the server. Bob needs a pull program. The client needs to pull the message from the server.



26. a) What is a Telnet? Write a detail note on it.

Ans: TELNET is an abbreviation for TErminaL NETwork. It is the standard TCP/IP protocol for virtual terminal service as proposed by ISO. Time-Sharing Environment TELNET was designed at a time when most operating systems, such as UNIX, were operating in a time-sharing environment. In such an environment, a large computer supports multiple users. The interaction between a user and the computer occurs through a terminal, which is usually a combination of keyboard, monitor, and mouse. Even a microcomputer can simulate a terminal with a terminal emulator.

Login: In a time-sharing environment, users are part of the system with some right to access resources. Each authorized user has an identification and probably a password. The user identification defines the user as part of the system. To access the system, the user logs into the system with a user id or login name. The system also includes password checking to prevent an unauthorized user from accessing the resources. **Local Login:** When a user logs into a local time-sharing system, it is called local login. As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver. The terminal driver passes the characters to the operating system. The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility

Remote Login When a user wants to access an application program or utility located on a remote machine, he or she performs remote login. Here the TELNET client and server programs come into use. The user sends the keystrokes to the terminal driver

where the local operating system accepts the characters but does not interpret them. The characters are sent to the TELNET client, which transforms the characters to a universal character set called Network Virtual Terminal (NVT) characters and delivers them to the local TCP/IP stack. Network Virtual Terminal (NVT) The mechanism to access a remote computer is complex. This is because every computer and its operating system accepts a special combination of characters as tokens. For example, the end-of-file token in a computer running the DOS operating system is Ctrl+z, while the UNIX operating system recognizes Ctrl+d. We are dealing with heterogeneous systems. If we want to access any remote computer in the world, we must first know what type of computer we will be connected to, and we must also install the specific terminal emulator used by that computer. TELNET solves this problem by defining a universal interface called the Network Virtual Terminal (NVT) character set. Via this interface, the client TELNET translates characters (data or commands) that come from the local terminal into NVT form and delivers them to the network. The server TELNET, on the other hand, translates data and commands from NVT form into the form acceptable by the remote computer NVT Character Set NVT uses two sets of characters, one for data and one for control. Both are 8-bit bytes

Format of data and control characters

a. Data Character b. Control Character

Data Characters: For data, NVT normally uses what is called NVT ASCII. This is an 8-bit character set in which the seven lowest order bits are the same as US ASCII and the highest order bit is 0.

Control Characters: To send control characters between computers (from client to server or vice versa), NVT uses an 8-bit character set in which the highest order bit is set to 1

Options: TELNET lets the client and server negotiate options before or during the use of the service. Options are extra features available to a user with a more sophisticated terminal. Users with simpler terminals can use default features. Some control characters discussed previously are used to define options.

The option descriptions are as follows:

Binary. This option allows the receiver to interpret every 8-bit character received, except IAC, as binary data. When IAC is received, the next character or characters are

interpreted as commands. However, if two consecutive IAC characters are received, the first is discarded and the second is interpreted as data.

Echo. This option allows the server to echo data received from the client. This means that every character sent by the client to the sender will be echoed back to the screen of the client terminal. In this case, the user terminal usually does not echo characters when they are typed but waits until it receives them from the server.

Suppress go-ahead. This option suppresses the go-ahead (GA) character (see section on Modes of Operation).

Status. This option allows the user or the process running on the client machine to get the status of the options being enabled at the server site.

Timing mark. This option allows one party to issue a timing mark that indicates all previously received data has been processed.

Terminal type. This option allows the client to send its terminal type.

Terminal speed. This option allows the client to send its terminal speed.

Line mode. This option allows the client to switch to the line mode

(OR)

b) Discuss in detail about VPN.

Ans:

VPN

A virtual private network (VPN) is a technology that creates a safe and encrypted connection over a less secure network, such as the internet. VPN technology was developed as a way to allow remote users and branch offices to securely access corporate applications and other resources. To ensure safety, data travels through secure tunnels, and VPN users must use authentication methods -- including passwords, tokens or other unique identification procedures -- to gain access to the VPN server. VPNs are used by remote workers who need access to corporate resources, consumers who may want to download files and business travelers who may want to log into sites that are geographically restricted. VPN services are critical conduits through which data can be transported safely and securely.

A virtual private network (VPN) is a technology that creates a safe and encrypted connection over a less secure network, such as the internet. VPN technology was developed as a way to allow remote users and branch offices to securely access corporate applications and other resources. To ensure safety, data travels through secure tunnels, and VPN users must use authentication methods --

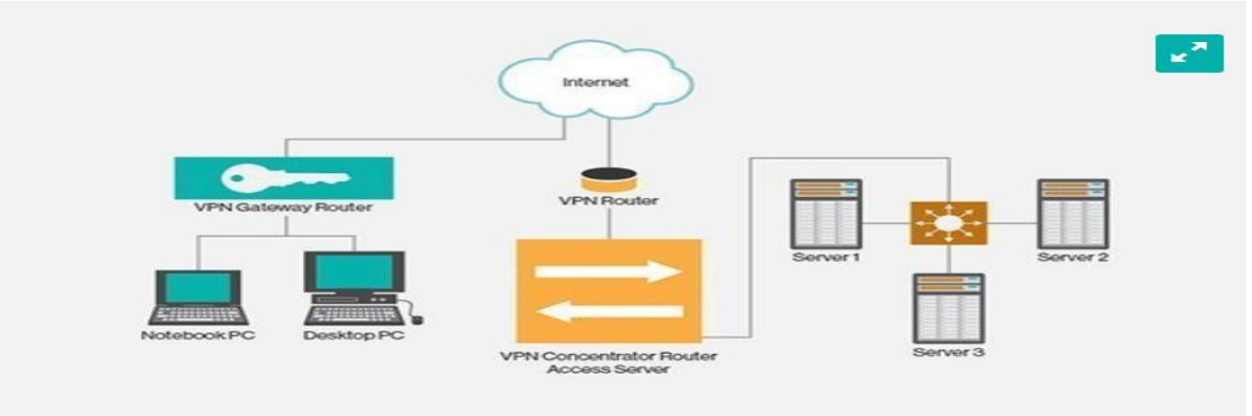
including passwords, tokens or other unique identification procedures -- to gain access to the VPN server. VPNs are used by remote workers who need access to corporate resources, consumers who may want to download files and business travelers who may want to log into sites that are geographically restricted. VPN services are critical conduits through which data can be transported safely and securely.

How a VPN works and why you should use one?

The two most common types of VPNs are remote access VPNs and site-to-site VPNs. A remote access VPN uses a public telecommunication infrastructure like the internet to provide remote users with secure access to their organization's network. This is especially important when employees are using a public Wi-Fi hotspot or other avenues to access the internet and connect to their corporate network. A VPN client on a remote user's computer or mobile device connects to a VPN gateway on the organization's network. The gateway typically requires the device to authenticate its identity. Then, it creates a network link back to the device that allows it to reach internal network resources -- e.g., file servers, printers and intranets -- as though the gateway is on the network locally.

A remote-access VPN usually relies on either IP Security (IPsec) or Secure Sockets Layer (SSL) to secure the connection, although SSL VPNs are often focused on supplying secure access to a single application rather than to the entire internal network. Some VPNs provide Layer 2 access to the target network; these require a tunneling protocol like the Point-to-Point Tunneling Protocol or the Layer 2 Tunneling Protocol running across the base IPsec connection. In addition to IPsec and SSL, other protocols used to secure VPN connectivity and encrypt data are Transport Layer Security and OpenVPN.

A site-to-site VPN uses a gateway device to connect an entire network in one location to a network in another -- usually a small branch connecting to a data center. End-node devices in the remote location do not need VPN clients because the gateway handles the connection.



Most site-to-site VPNs connecting over the internet use IPsec. It is also common for them to use carrier MPLS clouds rather than the public internet as the transport for site-to-site VPNs. Here, too, it is possible to have either Layer 3 connectivity (MPLS IP VPN) or Layer 2 (virtual private LAN service) running across the base transport. VPN services can also be defined as connections between specific computers, typically servers in separate data centers, when security requirements for their exchanges exceed what the enterprise network can deliver. Increasingly, enterprises also use VPN connections in either remote access mode or site-to-site mode to connect -- or connect to -- resources in a public infrastructure-as-a-service environment. Newer hybrid-access scenarios put the VPN gateway itself in the cloud, with a secure link from the cloud service provider into the internal network.

Benefits of using a VPN

The justification for using VPN access instead of a private network usually boils down to cost and feasibility: It is either not feasible to have a private network -- e.g., for a traveling sales rep -- or it is too costly to do so.