KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed University Established Under Section 3 of UGC Act 1956)
Coimbatore - 641021.
(For the candidates admitted from 2018 onwards)
Department of Computer Science, Applications & Information Technology

SUBJECT : **INTERNETWORKING WITH TCP/IP**
SEMESTER : II **L T P C**
SUBJECT CODE: **18CSP201** **CLASS** : **I M.Sc .CS** **4 0 0 4**

COURSE OBJECTIVE

- To understand about subnets using IP classes .
- To understand the key features and functions of TCP .
- To understand how basic routing works including the use of routing protocols.
- To understand about DNS and its applications
- To understand the concepts of Remote Login and VPN

COURSE OUTCOME

At the completion of the course, students will:

- Have the ability to analyze and differentiate networking protocols used in TCP/IP protocol suite.
- Understand IP Addressing Fundamentals
- Understand IPv4 forwarding and routing.
- Learn about host name resolution and the Domain Name System (DNS).
- Learn about services and operations of DHCP Servers and Domain Name Servers
- To create major applications using the key TCP/IP protocols
- To compare and contrast IP routing protocols

UNIT-I

Introduction: WAN WAN technologies - Protocols and Standards - TCP/IP protocol suite - Internetworking Devices - Classful IP Addressing – Subnetting – Supernetting – Classless Addressing

UNIT-II

ARP & RARP – Proxy ARP – ARP over ATM – ARP and RARP Protocol Format. IP Datagram – Fragmentation – Options – IP Datagram Format – Routing IP Datagrams – Checksum. ICMP – Types of Messages - Message Format – Error Reporting – Query – Checksum.

UNIT-III

Unicast Routing Protocol: Intra Domain and Inter Domain Routing – Distance Vector Routing – RIP – Link State Routing – OSPF – Path Vector Routing – BGP – Multicast Routing – Multicast

Routing Protocols. 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. (2009). TCP/IP Protocol Suite (3rd ed.). New Delhi: Tata McGraw Hill Publication.
   (Page Nos: 2-5 6-38 69-74 84-95 102-121 160-188 191-1-201 221-232 238-241 256-279 299-304 386-430 441-444 457-464 471-488 519-542 561-566 575-576 621-632 637-644 680-682)
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.
5. William Stallings. (2007). Data and Communication Network(8th ed.). New Delhi: Tata McGraw Hill.

WEB SITES

1. en.wikipedia.org/wiki/Internet_protocol_suite
2. http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies
3. www.yale.edu/pclt/COMM/TCPIP.HTM
4. www.w3schools.com/tcpip/default.asp

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

Lecture Plan

Subject Name: Internetworking With TCP/IP **Subject Code: 18CSP201**

Semester: II **Class: I M.Sc. CS**

Staff: Dr.S.Manju Priya

S.No	Topics	No. of Periods Required	Reference Materials
	Unit-I		
1	Introduction: WAN, WAN technologies	1hr	W2
2	Protocols and Standards	1hr	SR1:6-7
3	TCP/IP protocol suite	1hr	SR1:30-32
4	Internetworking Devices	1hr	SR1:69-74
5	Classful IP Addressing	1hr	SR1:84-95
6	Subnetting , Supernetting	1hr	SR1:102-125
7	Classless Addressing	1hr	SR5: 135-142
8	Recapitulation and Discussion of Important Questions	1hr	
	Total Hours	**8 hrs**	

Suggested Readings

SR1: Behrouz, A. Forouzan. (2009). TCP/IP Protocol Suite (3rd ed.). New Delhi: Tata McGraw Hill Publication.

SR2: Andrews, S. Tanenbaum. (2003). Computer Networks (4th ed.). New Delhi:Prentice Hall of India Private Ltd..

SR3: Buck Graham. (2007). TCP/IP Addressing (2nd ed.). New Delhi: Harcount India Private Limited.

SR4: Douglas, E. Comer. (2000). Computer Networks and Internets (4th ed.). New Delhi: Pearson Education.
SR5: William Stallings. (2007). Data and Communication Network(8th ed.). New Delhi: Tata McGraw Hill.
SR6: Kevin R. Fall, W. Richard Stevens, 2014, TCP/IP Illustrated, Volume 1: The Protocols, Addison Wesley, 2nd Edition.
SR7: Douglas E. Comer, 2013, Internetworking with TCP / IP: Principles, Protocols, and Architecture, PHI Learning Publications, 6th Edition.

Web References

W1: en.wikipedia.org/wiki/Internet_protocol_suite
W2: http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies
W3: www.yale.edu/pclt/comm/tcpip.ht
W4: www.w3schools.com/tcpip

Unit-II

1	**ARP & RARP** :Proxy ARP	1hr	SR1:166
2	ARP over ATM	1hr	SR1:166
3	ARP and RARP Protocol Format	1hr	SR1:173-175
4	IP Datagram , Fragmentation	1hr	SR1:180-188
5	Options, IP Datagram Format	1hr	SR1:191-200
6	Routing IP Datagrams , Checksum	1hr	SR1:200-202
7	ICMP, Types of Messages	1hr	SR1:212
8	Message Format	1hr	SR1:213
9	Error Reporting , Query , Checksum	1hr	SR1:213-227
10	Recapitulation and Discussion of Important Questions	1hr	-
	Total Hours	**10 hrs**	

Suggested Readings

SR1: Behrouz, A. Forouzan. (2009). TCP/IP Protocol Suite (3rd ed.). New Delhi: Tata McGraw Hill Publication.

SR2: Andrews, S. Tanenbaum. (2003). Computer Networks (4th ed.). New Delhi:Prentice Hall of India Private Ltd..
SR3: Buck Graham. (2007). TCP/IP Addressing (2nd ed.). New Delhi: Harcount India Private Limited.
SR4: Douglas, E. Comer. (2000). Computer Networks and Internets (4th ed.). New Delhi: Pearson Education.
SR5: William Stallings. (2007). Data and Communication Network(8th ed.). New Delhi: Tata McGraw Hill.
SR6: Kevin R. Fall, W. Richard Stevens, 2014, TCP/IP Illustrated, Volume 1: The Protocols, Addison Wesley, 2nd Edition.
SR7: Douglas E. Comer, 2013, Internetworking with TCP / IP: Principles, Protocols, and Architecture, PHI Learning Publications, 6th Edition.

Web References

W1: en.wikipedia.org/wiki/Internet_protocol_suite
W2: http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies
W3: www.yale.edu/pclt/comm/tcpip.ht
W4: www.w3schools.com/tcpip

Unit - III

1	Unicast Routing Protocol: Intra Domain and Inter Domain Routing	1hr	SR1:385-386
2	Distance Vector Routing	1hr	SR1: 387-391
3	RIP ,Link State Routing	1hr	SR1:392-403
4	OSPF	1hr	SR1:404-421
5	Path Vector Routing , BGP	1hr	SR1:421-430
6	Multicast Routing – Multicast Routing Protocols	1hr	SR1:441-444
7	Group Management , IGMP Message , IGMP Operation	1hr	SR1:237-243
8	Process to Process Communication , UDP Operation	1hr	SR1:256-267
9	TCP Services , Flow Control	1hr	SR1:276-305
10	Recapitulation and Discussion of Important Questions	1hr	-
	Total Hours	**10 hrs**	

Suggested Readings

SR1: Behrouz, A. Forouzan. (2009). TCP/IP Protocol Suite (3rd ed.). New Delhi: Tata McGraw Hill Publication.
SR2: Andrews, S. Tanenbaum. (2003). Computer Networks (4th ed.). New Delhi:Prentice

Hall of India Private Ltd..
SR3: Buck Graham. (2007). TCP/IP Addressing (2nd ed.). New Delhi: Harcount India Private Limited.
SR4: Douglas, E. Comer. (2000). Computer Networks and Internets (4th ed.). New Delhi: Pearson Education.
SR5: William Stallings. (2007). Data and Communication Network(8th ed.). New Delhi: Tata McGraw Hill.
SR6: Kevin R. Fall, W. Richard Stevens, 2014, TCP/IP Illustrated, Volume 1: The Protocols, Addison Wesley, 2nd Edition.
SR7: Douglas E. Comer, 2013, Internetworking with TCP / IP: Principles, Protocols, and Architecture, PHI Learning Publications, 6th Edition.

Web References

W1: en.wikipedia.org/wiki/Internet_protocol_suite
W2: http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies
W3: www.yale.edu/pclt/comm/tcpip.ht
W4: www.w3schools.com/tcpip

UNIT-IV			
1	BOOTP	1hr	SR1:457-462
2	DHCP , Address Discovery and Binding	1hr	SR1:463-467
3	DNS ,Name Space ,DNS in Internet	1hr	SR1:471-481
4	Resolution	1hr	SR1:481-482
5	Resource Records	1hr	SR2:624-628
6	Recapitulation and Discussion of Important Questions	1hr	
	Total Hours	**6 hrs**	

Suggested Readings

SR1: Behrouz, A. Forouzan. (2009). TCP/IP Protocol Suite (3rd ed.). New Delhi: Tata McGraw Hill Publication.
SR2: Andrews, S. Tanenbaum. (2003). Computer Networks (4th ed.). New Delhi:Prentice Hall of India Private Ltd..
SR3: Buck Graham. (2007). TCP/IP Addressing (2nd ed.). New Delhi: Harcount India Private Limited.
SR4: Douglas, E. Comer. (2000). Computer Networks and Internets (4th ed.). New Delhi: Pearson Education.
SR5: William Stallings. (2007). Data and Communication Network(8th ed.). New Delhi: Tata McGraw Hill.
SR6: Kevin R. Fall, W. Richard Stevens, 2014, TCP/IP Illustrated, Volume 1: The Protocols, Addison Wesley, 2nd Edition.
SR7: Douglas E. Comer, 2013, Internetworking with TCP / IP: Principles, Protocols, and

Architecture, PHI Learning Publications, 6th Edition.

Web References

W1: en.wikipedia.org/wiki/Internet_protocol_suite
W2: http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies
W3: www.yale.edu/pclt/comm/tcpip.ht
W4: www.w3schools.com/tcpip

UNIT-V

1	Remote Login, FTP	1hr	R5: 611-624 SR1:519-532
2	SMTP , SNMP	1hr	SR1:561-566 SR1:589-591
3	IP over ATM, WAN, Cells	1hr	R1:631-642 SR1:621- 626
4	Routing the Cells	1hr	SR1:626-627
5	ATMARP , Logical IP Subnets	1hr	SR1:627-633
6	VPN	1hr	SR1:680-682
7	Recapitulation and Discussion of Important Questions	1hr	
8	Discussion of Previous ESE Question Papers	1hr	
9	Discussion of Previous ESE Question Papers	1hr	
10	Discussion of Previous ESE Question Papers	1hr	
	Total Hours	**10 hrs**	
	Total Number of periods (8hr+9hr+11hr+7hr+13hr)	**44hrs**	

Suggested Readings

SR1: Behrouz, A. Forouzan. (2009). TCP/IP Protocol Suite (3rd ed.). New Delhi: Tata McGraw Hill Publication.
SR2: Andrews, S. Tanenbaum. (2003). Computer Networks (4th ed.). New Delhi:Prentice Hall of India Private Ltd..
SR3: Buck Graham. (2007). TCP/IP Addressing (2nd ed.). New Delhi: Harcount India Private Limited.
SR4: Douglas, E. Comer. (2000). Computer Networks and Internets (4th ed.). New Delhi: Pearson Education.
SR5: William Stallings. (2007). Data and Communication Network(8th ed.). New Delhi: Tata McGraw Hill.
SR6: Kevin R. Fall, W. Richard Stevens, 2014, TCP/IP Illustrated, Volume 1: The Protocols, Addison Wesley, 2nd Edition.
SR7: Douglas E. Comer, 2013, Internetworking with TCP / IP: Principles, Protocols, and Architecture, PHI Learning Publications, 6th Edition.

Web References

W1: en.wikipedia.org/wiki/Internet_protocol_suite
W2: http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies
W3: www.yale.edu/pclt/comm/tcpip.ht
W4: www.w3schools.com/tcpip

UNIT-I

SYLLABUS

Introduction: WAN - WAN technologies - Protocols and Standards - TCP/IP protocol suite - Internetworking Devices - Classful IP Addressing – Subnetting – Supernetting – Classless Addressing

Introduction:

What Is a WAN?

A WAN is a data communications network that covers a relatively broad geographic area and often uses transmission facilities provided by common carriers, such as telephone companies. WAN technologies function at the lower three layers of the OSI reference model: the physical layer, the data link layer, and the network layer.

WANs are all about exchanging information across wide geographic areas. Although the nature of a WAN—a network reliant on communications for covering sometimes vast distances—generally dictates slower throughput, longer delays, and a greater number of errors than typically occur on a LAN, a WAN is also the fastest, most effective means of transferring computer-based information currently available.

WAN technologies

A network is defined as a group of two or more computer systems linked together. There are many types of computer networks, including the following:

Local-area networks (LANs): The computers are geographically close together (that is, in the same building).

Wide-area networks (WANs): The computers are farther apart and are connected by telephone lines or radio waves.

Campus-area networks (CANs): The computers are within a limited geographic area, such as a campus or military base.

Metropolitan-area networks (MANs): A data network designed for a town or city.

Home-area networks (HANs): A network contained within a user's home that connects a person's digital devices.

ARPANET

ARPANET was the network that became the basis for the Internet. Based on a concept first published in 1967, ARPANET was developed under the direction of the U.S. Advanced Research Projects Agency (ARPA). In 1969, the idea became a modest reality with the interconnection of four university computers. The initial purpose was to communicate with and share computer resources among mainly scientific users at the connected institutions. ARPANET took advantage of the new idea of sending information in small units called packets that could be routed on different paths and reconstructed at their destination. The development of the TCP/IP protocols in the 1970s made it possible to expand the size of the network, which now had become a network of networks, in an orderly way.

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 (TCP) and Internet Protocol (IP) are two distinct computer network protocols. A protocol is an agreed-upon set of procedures and rules. When two computers follow the same protocols—the same set of rules—they can understand each other

and exchange data. TCP and IP are so commonly used together, however, that TCP/IP has become standard terminology for referring to this suite of protocols.

Transmission Control Protocol divides a message or file into packets that are transmitted over the internet and then reassembled when they reach their destination. Internet Protocol is responsible for the address of each packet so it is sent to the correct destination.

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

The **Computer Science Network** (**CSNET**) was a computer network that began operation in 1981 in the United States. Its purpose was to extend networking benefits, for computer science departments at academic and research institutions that could not be directly connected to ARPANET, due to funding or authorization limitations. It played a significant role in spreading awareness of, and access to, national networking and was a major milestone on the path to development of the global Internet.

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

NSFNET was a network for research computing deployed in the mid-1980s that in time also became the first backbone infrastructure for the commercial public Internet. Created as a result of a 1985 National Science Foundation (NSF) initiative, NSFNET established a high-speed connection among the five NSF supercomputer centers and the National Center for Atmospheric Research, and provided external access for scientists, researchers, and engineers who were not located near the computing centers.

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 ANSNE

ISP - Internet service provider

***I**nternet **S**ervice **P**rovider*, it refers to a company that provides Internet services, including personal and business access to the Internet. For a monthly fee, the service provider usually provides a software package, username, password and access phone number. Equipped with a modem, you can then log on to the Internet and browse the World Wide Web and USENET, and send and receive e-mail. For broadband access you typically receive the broadband modem hardware or pay a monthly fee for this equipment that is added to your ISP account billing.

In addition to serving individuals, ISPs also serve large companies, providing a direct connection from the company's networks to the Internet. ISPs themselves are connected to one another through *Network Access Points (NAPs)*. ISPs may also be called *IAPs (Internet Access Providers)*.

World Wide Web (WWW)

The World Wide Web (WWW) is combination of all resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP).The web was invented at CERN by Tim Berners-Lee. This invention has added the commercial applications to the Internet.

The following is a list of important Internet events in chronological order:

1. 1969. Four-node ARPANET established.
2. 1970. ARPA hosts implement NCP.
3. 1973. Development of TCP/IP suite begins.
4. 1977. An internet tested using TCP/IP.
5. 1978. UNIX distributed to academic/research sites.
6. 1981. CSNET established.
7. 1983. TCP/IP becomes the official protocol for ARPANET
8. 1983. MILNET was born.
9. 1986. NSFNET established
10. 1990. ARPANET decommissioned and replaced by NSFNET.
11. 1995. NSFNET goes back to being a research network.
12. 1995. Companies known as Internet Service Providers (ISPs) started.

Protocols and Standards

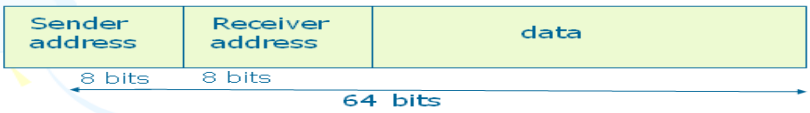
Protocol is a set of rules that govern all aspect of data communication between computers on a network. These rules include guidelines that regulate the following characteristics of a network: access method, allowed physical topologies, types of cabling, and speed of data transfer. A

protocol defines what, how, when it communicated. The key elements of a protocol are syntax, semantics and timing.

1. **Syntax**

The structure or format of the data.

Eg. A simple protocol;



2. **Semantics**

It refers to the meaning of each section of bits. It is how a particular pattern to be interpreted, and what action is to be taken based on that interpretation.

Eg. Does an address identify the route to be taken or the final of the message?

3. **Timing**

It refers to two characteristics:

a. When data to be sent
b. How fast it can be sent

Eg. If a sender produces data at 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 communication.

Data communication standards fall into two categories: *de facto* (meaning "by fact" or "by convention") and *de jure* (meaning "by law" and "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.
- **De jure**. De jure standards are those that have been legislated by an officially recognized body.

TCP/IP protocol suite

TCP/IP (Transmission Control Protocol/Internet Protocol)

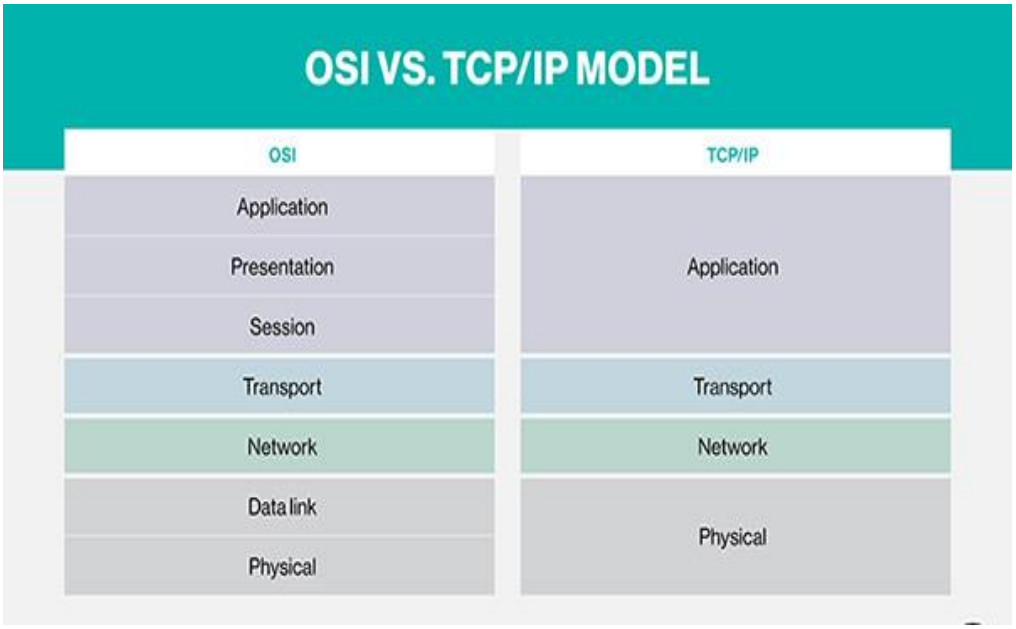
TCP/IP, or the Transmission Control Protocol/Internet Protocol, is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP can also be used as a communications protocol in a private network (an intranet or an extranet). The entire internet protocol suite is a set of rules and procedures -- is commonly referred to as TCP/IP, though others are included in the suite.

TCP/IP specifies how data is exchanged over the internet by providing end-to-end communications that identify how it should be broken into packets, addressed, transmitted, routed and received at the destination. TCP/IP requires little central management, and it is designed to make networks reliable, with the ability to recover automatically from the failure of any device on the network.

The two main protocols in the internet protocol suite serve specific functions. TCP defines how applications can create channels of communication across a network. It also manages how a message is assembled into smaller packets before they are then transmitted over the internet and reassembled in the right order at the destination address.

IP defines how to address and route each packet to make sure it reaches the right destination. Each gateway computer on the network checks this IP address to determine where to forward the message.

The transport layer itself, however, is stateful. It transmits a single message, and its connection remains in place until all the packets in a message have been received and reassembled at the destination.



The TCP/IP model differs slightly from the seven-layer Open Systems Interconnection (OSI) networking model designed after it, which defines how applications can communicate over a network.

TCP/IP model layers

TCP/IP functionality is divided into four layers, each of which includes specific protocols.

The **application layer** provides applications with standardized data exchange. Its protocols include the Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Post Office

Protocol 3 (POP3), Simple Mail Transfer Protocol (SMTP) and Simple Network Management Protocol (SNMP).

The **transport layer** is responsible for maintaining end-to-end communications across the network. TCP handles communications between hosts and provides flow control, multiplexing and reliability. The transport protocols include TCP and User Datagram Protocol (UDP), which is sometimes used instead of TCP for special purposes.

The **network layer**, also called the internet layer, deals with packets and connects independent networks to transport the packets across network boundaries. The network layer protocols are the IP and the Internet Control Message Protocol (ICMP), which is used for error reporting.

The **physical layer** consists of protocols that operate only on a link -- the network component that interconnects nodes or hosts in the network. The protocols in this layer include Ethernet for local area networks (LANs) and the Address Resolution Protocol (ARP).

Advantages of TCP/IP

TCP/IP is nonproprietary and, as a result, is not controlled by any single company. Therefore, the internet protocol suite can be modified easily. It is compatible with all operating systems, so it can communicate with any other system. The internet protocol suite is also compatible with all types of computer hardware and networks.

Internetworking 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.

Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to

which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do no amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

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.

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

One may ask what is the difference in functionality between a bridge and a repeater. 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. A bridge has a table used in filtering decisions.

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,

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.

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.

Three-LayerSwitch

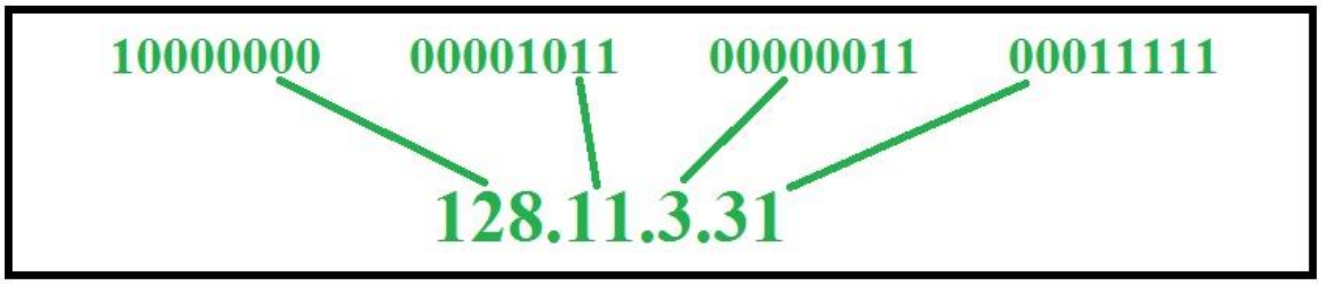
A three-layer switch is a router; a router with an improved design to allow better performance. A three-layer switch can receive, process, and dispatch a packet much faster than a traditional

router even though the functionality is the same. In this book, to avoid confusion, we use the term router for a three-layer switch.

Classful IP Addressing

IP address is an address having information about how to reach a specific host, especially outside the LAN. An IP address is a 32 bit unique address having an address space of 2^{32}. Generally, there are two notations in which IP address is written, dotted decimal notation and hexadecimal notation.

Dotted Decimal Notation



Hexadecimal Notation

Some points to be noted about dotted decimal notation :
1. The value of any segment (byte) is between 0 and 255 (both included).
2. There are no zeroes preceding the value in any segment (054 is wrong, 54 is correct).

Classful Addressing

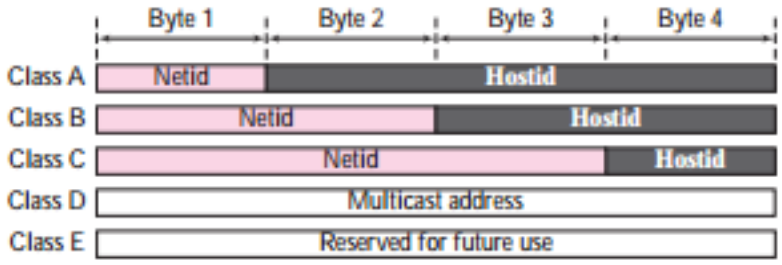
The 32 bit IP address is divided into five sub-classes. These are:

- Class A
- Class B
- Class C
- Class D
- Class E

Each of these classes has a valid range of IP addresses. Classes D and E are reserved for multicast and experimental purposes respectively. The orders of bits in the first octet determine the classes of IP address.

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.

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.

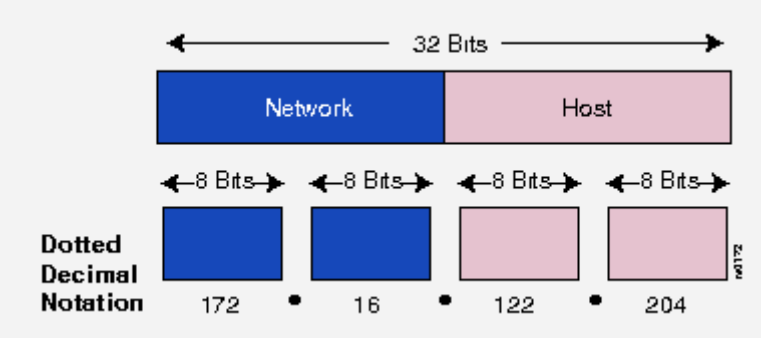
Recognizing Classes

We can find the class of an address when the address is given either in binary or dotted decimal notation.

IP Address Format:

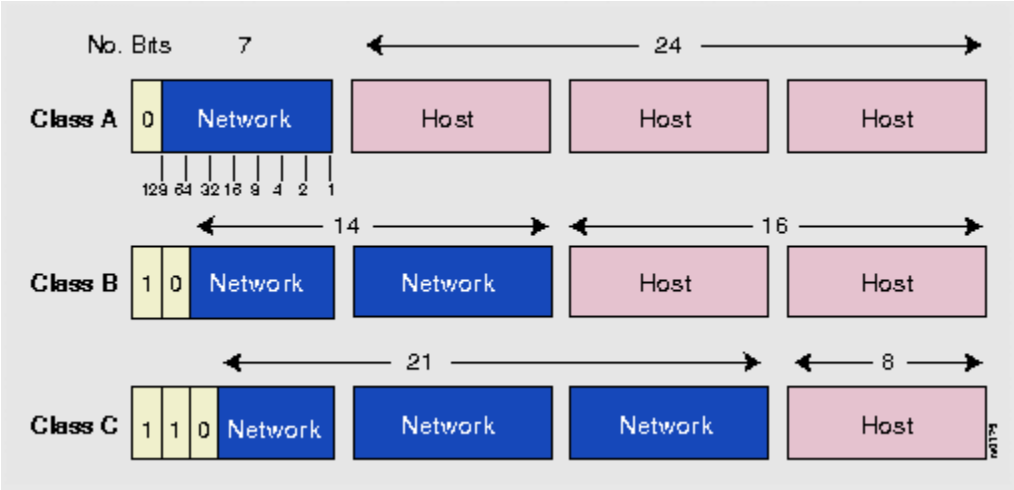
The 32-bit IP address is grouped 8 bits at a time, each group of 8 bits is an octet. Each of the four octets are separated by a dot, and represented in decimal format, this is known as dotted decimal notation. Each bit in an octet has a binary weight (128, 64, 32, 16, 8, 4, 2, 1). The minimum value for an octet is 0 (all bits set to 0), and the maximum value for an octet is 255 (all bits set to 1).

The following figure shows the basic format of a typical IP address:



IP Address Classes:

IP addressing supports three different commercial address classes; Class A, Class B, and Class C. The following figure summarizes the network and host portion of each address class:



In a class A address, the first octet is the network portion, so the class A address of, 10.1.25.1, has a major network address of 10. Octets 2, 3, and 4 (the next 24 bits) are for the hosts. Class A addresses are used for networks that have more than 65,536 hosts (actually, up to 16,581,375 hosts).

In a class B address, the first two octets are the network portion, so the class B address of, 172.16.122.204, has a major network address of 172.16. Octets 3 and 4 (the next 16 bits) are for the hosts. Class B addresses are used for networks that have between 256 and 65,536 hosts.

In a class C address, the first three octets are the network portion. The class C address of, 193.18.9.45, has a major network address of 193.18.9. Octet 4 (the last 8 bits) is for hosts. Class C addresses are used for networks with less than 254 hosts.

Example

Find the class of each address:

a. 227.12.14.87

b. 193.14.56.22

c. 14.23.120.8

d. 252.5.15.111

Solution

a. The first byte is 227 (between 224 and 239); the class is D.
b. The first byte is 193 (between 192 and 223); the class is C.
c. The first byte is 14 (between 0 and 127); the class is A.
d. The first byte is 252 (between 240 and 255); the class is E.

Determining the Class from the First-Octet:

The class of address can be easily determined by examining the first octet of the address, and mapping that value to a class range in the table below:

The left-most (high-order) bits in the first octet indicate the network class. For example, given an IP address of 172.31.1.2, the first octet is 172. 172 falls between 128 and 191, so 172.31.1.2 is a Class B address.

First octet	Address Class
0-127	Class A
128-191	Class B
192-223	Class C
224-239	Class D
240-255	Class E

Classful Network Masks:

Each of the commercial address classes has a set classful network mask. The network mask defines which bits out of the 32 bit of the address are defined as the network portion and which

are the host portion. The network mask is calculated by setting all bits to a value of 1 in the octets designated for the network portion and all bits to a value of 0 in the octets designated for the host portion.

As stated above, a Class A address has the first octet as the network portion and the remaining 3 octets as the host portion. Therefore, a Class A network mask is defined as 255.0.0.0.

A Class B address has the first and second octets as the network portion and the third and fourth octets as the host portion. A Class B network mask is shown as 255.255.0.0.

A Class C address has the first, second, and third octet as the network portion and the last octet as the host portion. A Class C network mask is shown as 255.255.255.0.

For example, 128.8.74.1 is a Class B address because the first octet, 128, lies in the 128-191 range. Likewise, 10.10.191.1 is a Class A address (because the first octet is 10) and 208.130.29.33 is a Class C (because the first octet is 208).

Example

From the 32 bit IP address we can create dotted decimal notation by converting each byte to a decimal number between 0 and 255. We can also identity the class of an IP address by observing the first few bits of 1st byte of each IP address. The various examples are given in the table below.

S.No.	IP address				Dotted decimal notation	Class
1.	11011101	10001111	11111100	11001111	221.143.252.207	ClassC
2.	10011101	10001111	11111100	11001111	157.143.252.207	Class B
3.	01111011	10001111	11111100	11001111	123.143.252.207	Class A
4.	11110101	10001111	11111100	11001111	245.143.252.207	Class E
5.	11101011	10001111	11111100	11001111	235.143.252.207	Class D

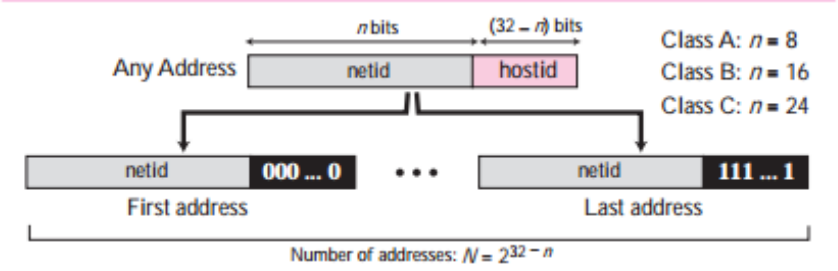
We can also find the network address i.e. net id and host id for the IP address (dotted decimal notation) with help of class as shown below.

S.No	IP Address	Class	Net id	Host id
1.	227.34.78.7	Class D	no net id	no host id
2.	4.23.145.90	Class A	4	23.145.90
3.	198.76.9.23	Class C	198.76.9	23
4.	129.6.8.4	Class B	129.6	8.4
5.	246.7.3.8	Class E	no net id	no host id

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. We can now find these three pieces of information as shown in the below figure.

1. The number of addresses in the block, N, can be found using $N = 2^{32-n}$
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 first address, we keep the n leftmost bits and set the (32-n) rightmost bits all to 1s"



Information extraction in classful addressing

Subnetting

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.

Subnetting enables the network administrator to further divide the host part of the address into two or more subnets. In this case, a part of the host address is reserved to identify the particular subnet. This is easier to see if we show the IP address in binary format.

The full address is:

10010110.11010111.00010001.00001001

The Class B network part is:

10010110.11010111

The host address is:

00010001.00001001

If this network is divided into 14 subnets, however, then the first 4 bits of the host address (0001) are reserved for identifying the subnet.

The subnet mask is the network address plus the bits reserved for identifying the subnetwork -- by convention, the bits for the network address are all set to 1, though it would also work if the bits were set exactly as in the network address. In this case, therefore, the subnet mask would be 11111111.11111111.11110000.00000000. It's called a *mask* because it can be used to identify the subnet to which an IP address belongs by performing a bitwise AND operation on the mask and the IP address. The result is the subnetwork address:

Subnet Mask	255.255.240.000	11111111.11111111.11110000.00000000
IP Address	150.215.017.009	10010110.11010111.00010001.00001001
Subnet Address	150.215.016.000	10010110.11010111.00010000.00000000

The subnet address, therefore, is 150.215.016.000.

For example how will we figure out network partition and host partition from IP address 192.168.1.10 ? Here we need subnet mask to get details about network address and host address.

- In decimal notation subnet mask value 1 to 255 represent network address and value 0 [Zero] represent host address.
- In binary notation subnet mask **ON** bit [1] represent network address while **OFF** bit[0] represent host address.

In decimal notation

IP address 192.168.1.10
Subnet mask 255.255.255.0

Network address is **192.168.1** and host address is **10**.

Examples

Now that you have an understanding of subnetting, put this knowledge to use. In this example, you are given two address / mask combinations, written with the prefix/length notation, which have been assigned to two devices. Your task is to determine if these devices are on the same subnet or different subnets. You can use the address and mask of each device in order to determine to which subnet each address belongs.

Device A: 172.16.17.30/20

Device B: 172.16.28.15/20

Determine the Subnet for DeviceA:

172.16.17.30 - 10101100.00010000.00010001.00011110

255.255.240.0 - 11111111.11111111.11110000.00000000

----------| sub|----------

subnet = 10101100.00010000.00010000.00000000 = 172.16.16.0

Looking at the address bits that have a corresponding mask bit set to one, and setting all the other address bits to zero (this is equivalent to performing a logical "AND" between the mask and address), shows you to which subnet this address belongs. In this case, DeviceA belongs to subnet 172.16.16.0.

Determine the Subnet for DeviceB:

172.16.28.15 - 10101100.00010000.00011100.00001111

255.255.240.0 - 11111111.11111111.11110000.00000000

----------| sub|----------

subnet = 10101100.00010000.00010000.00000000 = 172.16.16.0

From these determinations, DeviceA and DeviceB have addresses that are part of the same subnet.

Supernetting

Supernetting is the opposite of Subnetting. In subnetting, a single big network is divided into multiple smaller subnetworks. In Supernetting, multiple networks are combined into a bigger network termed as a Supernetwork or Supernet.

Supernetting is mainly used in Route Summarization, where routes to multiple networks with similar network prefixes are combined into a single routing entry, with the routing entry pointing to a Super network, encompassing all the networks. This in turn significantly reduces the size of routing tables and also the size of routing updates exchanged by routing protocols.

More specifically,

- When multiple networks are combined to form a bigger network, it is termed as super-netting
- Supernetting is used in route aggregation to reduce the size of routing tables and routing table updates

Examples for Supernetting

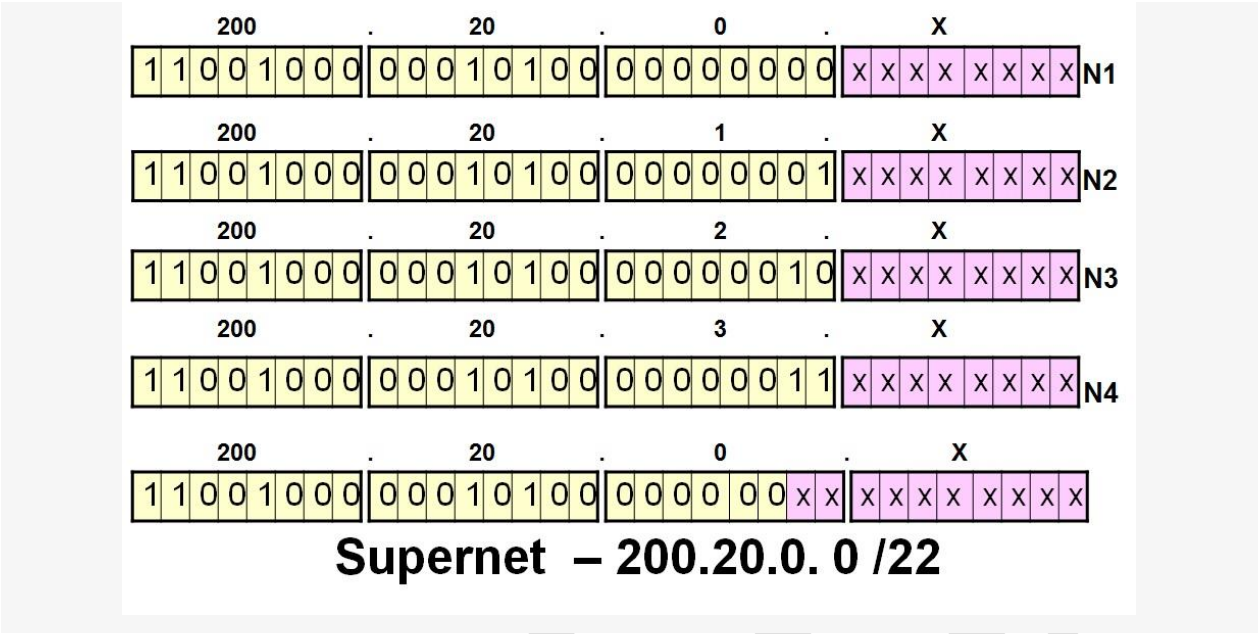
Example – 1

Consider two networks with addresses **200.20.0.x/24** and **200.20.1.x/24.** Both these networks have the same 23 bits network prefix and differ only in their 24th bit. Hence these two networks could be combined and summarized as a Super network with the address **200.20.0.x/23**.

Note that the network mask has been reduced to 23 for the supernet. While subnetting borrows bits from the host portion of the IP address, supernetting borrows bits from the network portion of the IP address.

Example – 2

Similarly, if you take the four networks **200.20.0.x/24, 200.20.1.x/24 , 200.20.2.x & 200.20.3.x/24,** they all have the same 22 bits network prefix and differ only starting from the 23rd bit. Hence these four networks could be combined into a single supernet as **200.20.0.x/22.** The diagram below illustrates this example of combining the four networks into a single Super network.



In the above example, all the four networks N1 to N4 have the same network prefix for the first 22 bits. Hence, Supernetting exploits this property and combines these four networks into a single Supernet, with a subnet mask of /22, for route aggregation purposes. Hence using this technique, four routing entries can be combined into a single routing entry, thereby reducing the size of routing tables and routing updates.

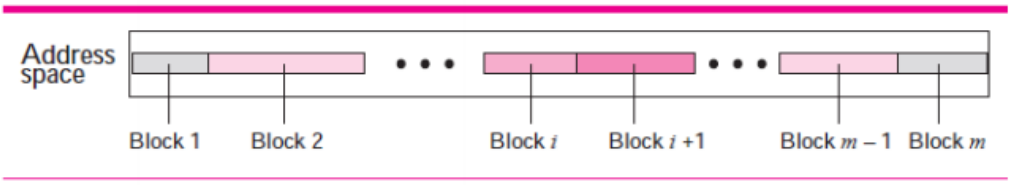
Classless Addressing

To reduce the wastage of IP addresses in a block, we use sub-netting. What we do is that we use host id bits as net id bits of a classful IP address. We give the IP address and define the number of bits for mask along with it (usually followed by a '/' symbol), like, 192.168.1.1/28. Here, subnet mask is found by putting the given number of bits out of 32 as 1, like, in the given address, we need to put 28 out of 32 bits as 1 and the rest as 0, and so, the subnet mask would be 255.255.255.240.

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. Although the long

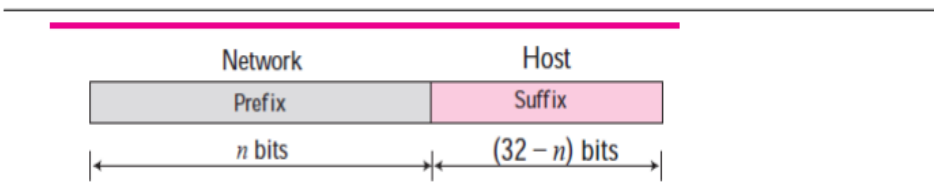
range solution has already been devised and is called IPv6 (see Chapters 26 to 28), a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called classless addressing.

Variable-Length Blocks In classless addressing, the whole address space is divided into variable length blocks. Theoretically, we can have a block of 20 , 21 , 22 , . . . , 2 32 addresses. The only restriction, as we discuss later, 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. Figure 5.27 shows the division of the whole address space into non overlapping blocks.



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 netid defined the network; the hostid defined the host in the network. 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. Figure 5.28 shows the prefix and suffix in a classless block.



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.

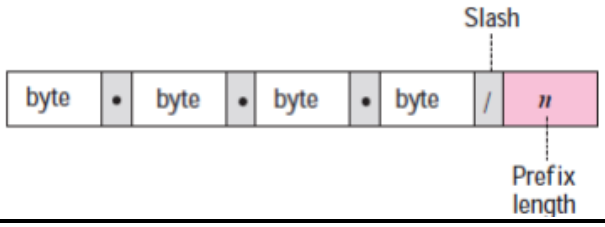
Example What is the prefix length and suffix length if the whole Internet is considered as one single block with 4,294,967,296 addresses? Solution In this case, the prefix length is 0 and the suffix length is 32. All 32 bits vary to define 232 = 4,294,967,296 hosts in this single block.

Example What is the prefix length and suffix length if the Internet is divided into 4,294,967,296 blocks and each block has one single address? Solution In this case, the prefix length for each block is 32 and the suffix length is 0. All 32 bits are needed to define 232 = 4,294,967,296 blocks. The only address in each block is defined by the block itself.

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 in the below figure.



Slash notation

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

Network Mask The idea of network mask in classless addressing is the same as the one in classful addressing. A network mask is a 32-bit number with the n leftmost bits all set to 0s and the rest of the bits all set to 1s.

Example The following addresses are defined using slash notations.

a. In the address 12.23.24.78/8, the network mask is 255.0.0.0. The mask has eight 1s and twenty-four 0s.

The prefix length is 8; the suffix length is 24.

b. In the address 130.11.232.156/16, the network mask is 255.255.0.0. The mask has sixteen 1s and sixteen 0s.

The prefix length is 16; the suffix length is 16.

c. In the address 167.199.170.82/27, the network mask is 255.255.255.224. The mask has twenty seven 1s and five 0s.
The prefix length is 27; the suffix length is 5.

Example One of the addresses in a block is 167.199.170.82/27. Find the number of addresses in the network, the first address, and the last address.

Solution The value of n is 27. The network mask has twenty-seven 1s and five 0s. It is 255.255.255.240.

a. The number of addresses in the network is 232 − n = 232 − n = 25 = 32.

b. We use the AND operation to find the first address (network address).The first address is 167.199.170.64/27.

c. To find the last address, we first find the complement of the n

Possible Questions

Two marks

1. Mention the different types of network topologies.
2. Define TCP.
3. What is CSNET?
4. What are standards?
5. What is the address space of a system with 8 bit and 16 bit addresses?
6. Change the IP addresses from dotted decimal notation to hexadecimal notation:

   (i) 114.34.2.8 (ii) 129.14.6.8
7. Find the class of the IP addresses:

   (i) 238.34.2.1 (ii) 241.34.2.8
8. Differentiate classful and classless addressing.
9. What is supernetting?
10. Mention some of internetworking device.

Eight Marks

1. Explain TCP/IP protocol suite with a sketch.
2. Discuss about subnetting in classful addressing.

3. Brief about any two connecting devices with a neat diagram.
4. Write about classless addressing with example.
5. Elaborate on supernetting in classful addressing.
6. Write about classes and blocks in Class A, B, C, D and E.
7. Write the procedures to find first address and last address in a block in classless addressing.
8. Discuss the importance of mask with an example in classful addressing.
9. Write about recognizing classes and masks in Classful Addressing.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

ONE MARK QUESTIONS

DEPARTMENT OF CS, CA & IT

STAFF NAME: S.MANJU PRIYA **SUBJECT NAME: INTERNETWORKING WITH TCP/IP**

SUB.CODE: 18CSP201 **UNIT I** **SEMESTER: II**

S.NO	Question	Choice1	Choice2	Choice3	Choice4	Ans
1	_____ is composed of hundreds of thousands of interconnected networks.	Internet	Intranet	Extranet	Arpanet	Internet
2	ARPA Stands for _____	Advanced Research Protocol	Automated Research Provider	Advanced Research Project	None	Advanced Research Project
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	Defund	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 Interconnectio	Organizationa l Standard interface	Open Source interconnectio n	Open System Interconnectio n
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	pplication Layer
15	In TCP/IP application layer is the combination of ____,_____ and ___	Application , Network, Data Link	Application , Network, Physical	Application , Network, Session	Application , Session , Presentation	Application , Session , Presentation
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	Regenerator, Resistance	Regenerator , Amplifier
18	_____ is a multi port repeater	Bridge	Router	Hub	switches	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	hub	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	ARP
30	_____is an internet work address	physical address	Logical address	IP address	Network address.	2
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	Permananet mapping	Dynamic mapping
33	RARP Stand for_____	Resolution Address Reverse	Routing Address Resolution	Routing Address Reverse	Reverse Address Resolution	Reverse Address Resolution
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	Permananet mapping	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	_____is used to create a subnetting effect.	PROXY ARP	ARP	RARP	ICMP	PROXY ARP
39	Packets in the IP Layer are called_____	Components	Interface	Tokens	Datagram.	Datagram.
40	The process of dividing the datagram to pass through the networks _____	Segmentation	Fragmentation	Splitting	Encapsulation	Fragmentation
41	The two-bit subfield defines the general purpose of the option is _____	Copy	Class	Number	object	Class
42	_____ program is used to find if a host is alive and responding.	Traceroute	Ping	Tracert	tracewindow	Ping
43	A parameter problem message can be created by a _____	Router	hub	Bridge	Both a and b	Router
44	_____table is used by the reassembly module.	Fragmentation table	Bridge table	Reassembly table	Routing table	Reassembly table
45	_____ is a local address.	physical	logical	network	IP	physical
46	_____ attaches to two or more physical networks and forwards IP datagrams	Router	Hub	Bridge	switch	Router
47	Internet routers could be portioned into _____ and _____	MILNET and ARPANET	Core and Non core	Individual and Non individual	separator	Core and Non core
48	_____ routers are controlled by individual routers	Core	Non core	empty	central	Non core

49	_____ travels from routers to routers	Text	Numbers	Datagrams	Characters.	Datagrams
50	In 1972, _____ Project was Started.	Internet	Inter network	Internetting	supernettting	Internetting
51	Other than ARPANET, the other two Networks are _____ and _____	Packet ratio & Packet satellite	Packet radio & Packet Mobile	Packet mobile & Packet	Packet switch , packet radio	Packet ratio & Packet satellite
52	_____ is a less expensive Network	CSNET	MILNET	ANSNET	NSFNET	CSNET
53	In NSFNET data is transferred at _____ rate.	1.439 Mbps	1.544Mbps	1.644 Mbps	1.744 Mbps	1.544Mbps
54	Repeaters and Hub Operate in _____ Layer of Internet Model	First	Second	Third	First & Second.	First
55	Repeater receives _____ and regenerates _____ pattern.	Data, bytes	Signal, Bit	Decimal, Signal	hexa, binary	Signal, Bit
56	Repeater connects _____ of a LAN.	Part	Mode	Mode2	Segment	Segment
57	Router connects _____ Lan to create _____	Segment, Network	Sector, Internet	Independent, Internetwork	dependent, sector	Independent, Internetwork
58	Router changes physical address into _____	Datagram	Packet	Signal	bytes	Packet
59	_____notation uses Dot for separating bytes	Decimal	Binary	Hexa	Octal	Decimal
60	Migration is very fast in _____ addressing	Classful addressing	Classless addressing	Both	supernetting	Classful addressing
61	If all the bits are one then it is a _____ address	Class A	Class B	Class C	Class E	Class E
62	A Block of address _____ is same.	First	Second	Third	Fourth.	First

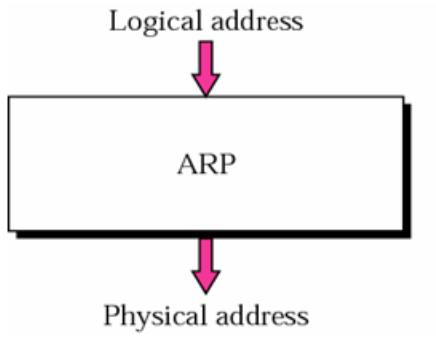
63	The First address in a block is _____	Net-id	Host – id	Subnet	Network address	Network address
64	The class of 208.34.55.12 is _____	Class A	Class B	Class C	Class D	Class C
65	The Net id of 114.34.2.8 IP Address is _____	114	114.34	114.34.2	14.43.28	114

UNIT-II

Introduction: ARP & RARP – Proxy ARP – ARP over ATM – ARP and RARP Protocol Format. IP Datagram – Fragmentation – Options – IP Datagram Format – Routing IP Datagrams – Checksum. ICMP – Types of Messages - Message Format – Error Reporting – Query – Checksum.

ARP

Address Resolution Protocol (ARP) is a protocol for mapping an Internet Protocol address (IP address) to a physical machine address that is recognized in the local network. For example, in IP Version 4, the most common level of IP in use today, an address is 32 bits long. In an Ethernet local area network, however, addresses for attached devices are 48 bits long. (The physical machine address is also known as a Media Access Control or MAC address.) A table, usually called the ARP cache, is used to maintain a correlation between each MAC address and its corresponding IP address. ARP provides the protocol rules for making this correlation and providing address conversion in both directions. 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.

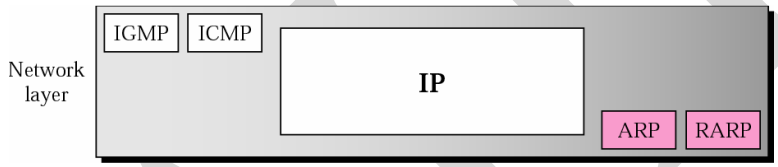


The term address resolution refers to the process of finding an address of a computer in a network. The address is "resolved" using a protocol in which a piece of information is sent by a

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.

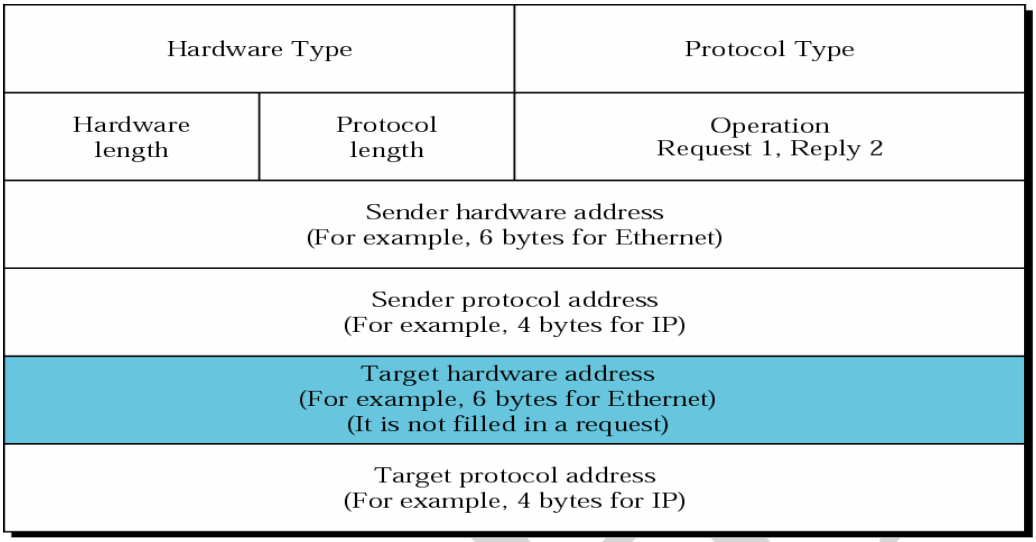


ARP Process

The Address Resolution itself is a two step process – a request and a response. It starts with the initiator sending an ARP Request as a broadcast frame to the entire network. This request *must* be a broadcast, because at this point the initiator does not know the target's MAC address, and is therefore unable to send a unicast frame to the target.

Since it was a broadcast, all nodes on the network will receive the ARP Request. All nodes will take a look at the content of the ARP request to determine whether they are the intended target. The nodes which are *not* the intended target will silently discard the packet. The node which *is* the target of the ARP Request will then send an ARP Response back to the original sender. Since the target knows who sent the initial ARP Request, it is able to send the ARP Response unicast, directly back to the initiator.

The **Packet format** for ARP is given below.



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.

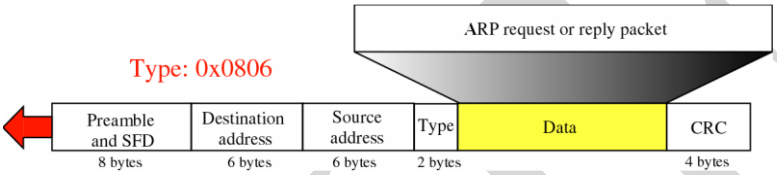
Sender hardware address (SHA): Hardware (MAC) address of the sender.

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.

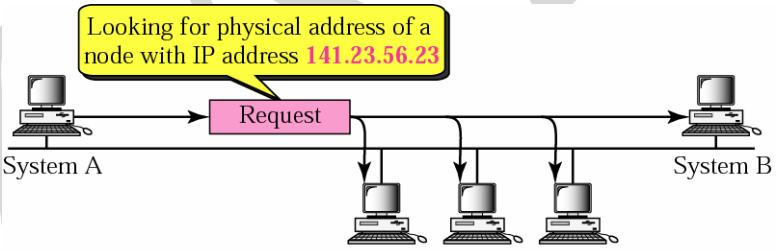
Target protocol address (TPA): Upper layer protocol address of the intended receiver.

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

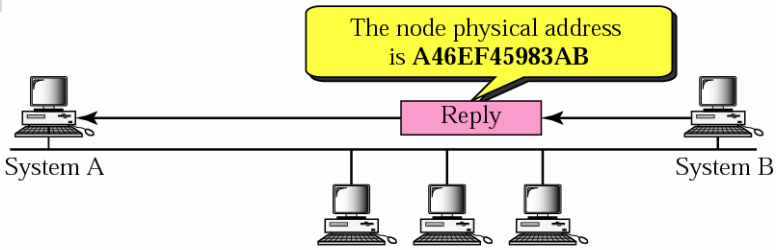


ARP operation

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

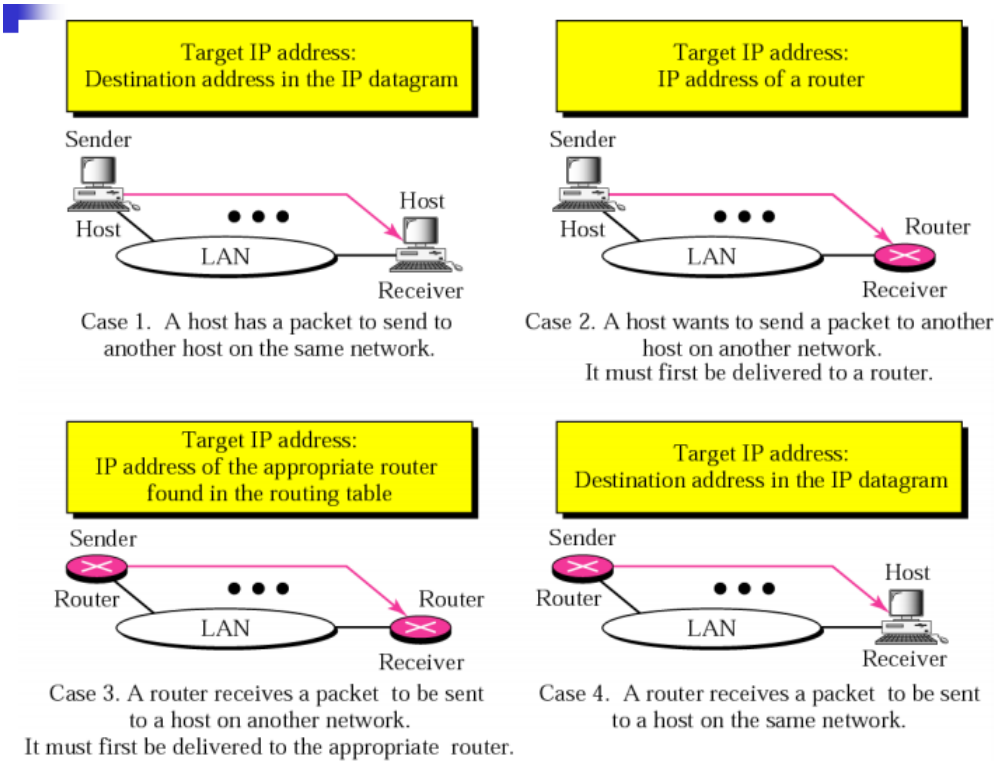


a. ARP request is broadcast



b. ARP reply is unicast

Four cases using ARP



Proxy ARP

ARP was designed to be used by devices that are directly connected on a local network. Each device on the network should be capable of sending both unicast and broadcast transmissions directly to each other one. Normally, if device *A* and device *B* are separated by a router, they would not be considered local to each other. Device *A* would not send directly to *B* or vice-versa; they would send to the router instead at layer two, and would be considered "two hops apart" at layer three.

Why Proxy ARP Is Needed

In contrast to the normal situation, in some networks there might be two physical network segments connected by a router that are in the same IP network or subnetwork. In other words, device *A* and device *B* might be on different networks at the data link layer level, but on the same

IP network or subnet. When this happens, *A* and *B* will each think the other is on the local network when they look to send IP datagrams.

In this situation, suppose that *A* wants to send a datagram to *B*. It doesn't have *B*'s hardware address in the cache, so it begins an address resolution. When it broadcasts the *ARP Request* message to get *B*'s hardware address, however, it will quickly run into a problem: *B* is in fact not on *A*'s local network. The router between them will not pass *A*'s broadcast onto *B*'s part of the network, because routers don't pass hardware-layer broadcasts. *B* will never get the request and thus *A* will not get a reply containing *B*'s hardware address.

Proxy ARP Operation

The solution to this situation is called *ARP proxying* or *Proxy ARP*. In this technique, the router that sits between the local networks is configured to respond to device *A*'s broadcast on behalf of device *B*. It does not send back to *A* the hardware address of device *B*; since they are not on the same network, *A* cannot send directly to *B* anyway. Instead, the router sends *A* its own hardware address. *A* then sends to the router, which forwards the message to *B* on the other network. Of course, the router also does the same thing on *A*'s behalf for *B*, and for every other device on both networks, when a broadcast is sent that targets a device not on the same actual physical network as the resolution initiator. This is illustrated in Figure .

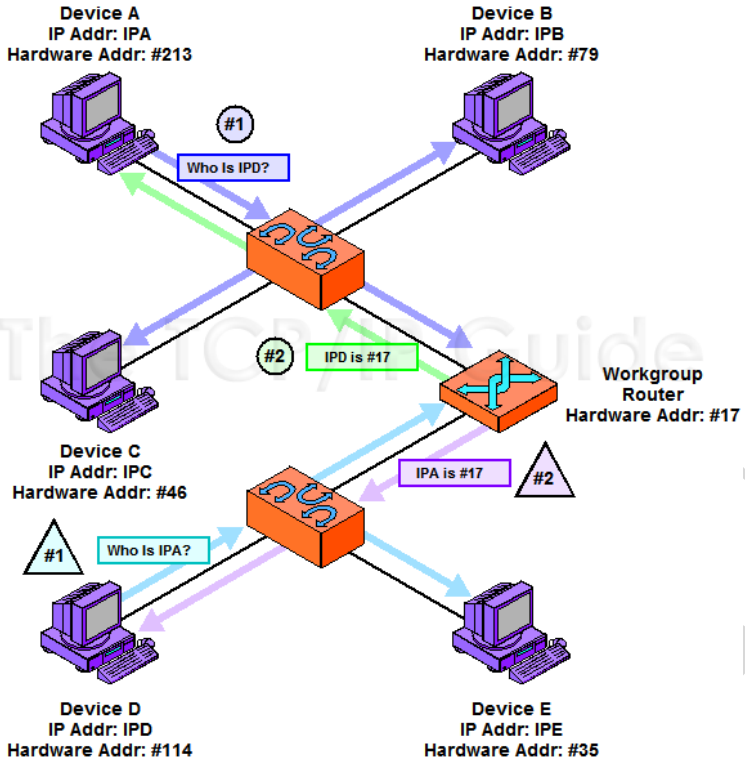


Figure ARP Proxy Operation

In this small internetwork, a single router connects two LANs that are on the same IP network or subnet. The router will not pass ARP broadcasts, but has been configured to act as an ARP proxy. In this example, device *A* and device *D* are each trying to send an IP datagram to the other, and so each broadcasts an *ARP Request*. The router responds to the request sent by Device *A* as if it were Device *D*, giving to *A* its own hardware address (without propagating Device *A*'s broadcast.) It will forward the message sent by*A* to *D* on *D*'s network. Similarly, it responds to Device *D* as if it were Device *A*, giving its own address, then forwarding what *D* sends to it over to the network where *A* is located.

Proxy ARP provides flexibility for networks where hosts are not all actually on the same physical network but are configured as if they were at the network layer. It can be used to provide support in other special situations where a device cannot respond directly to ARP

message broadcasts. It may be used when a firewall is configured for security purposes. A type of proxying is also used as part of the Mobile IP protocol, to solve the problem of address resolution when a mobile device travels away from its home network.

Advantages and Disadvantages of Proxying

The main advantage of proxying is that it is transparent to the hosts on the different physical network segments. The technique has some drawbacks though. First, it introduces added complexity. Second, if more than one router connects two physical networks using the same network ID, problems may arise. Third, it introduces potential security risks; since it essentially means that a router "impersonates" devices in acting as a proxy for them, raising the potential for a device to spoof another. For these reasons, it may be better to redesign the network so routing is done between physical networks separated by a router, if possible.

ARP over ATM

Classical IP and ARP over ATM standard specifies the mechanism for implementing Internet Protocol (IP) over ATM. Because ATM is connection-oriented technology and IP is a datagram-oriented technology, mapping the IP over ATM is not trivial.

In general, the ATM network is divided into logical IP subnetworks (LISs). Each LIS is comprised of some number of ATM stations. LISs are analogous to traditional LAN segments. LISs are interconnected using routers. A particular adapter (on an ATM station) can be part of multiple LISs. This feature can be very useful for implementing routers.

RFC1577 specifies RFC1483, which specifies logical link control/Sub-Network Access Protocol (LLC/SNAP) encapsulation as the default. In PVC networks for each IP station, all PVCs must be manually defined by configuring VPI:VCI values. If LLC/SNAP encapsulation is not being used, the destination IP address associated with each VPI:VCI must be defined.

For SVC networks, RFC1577 specifies an ARP server per LIS. The purpose of the ARP server is to resolve IP addresses into ATM addresses without using broadcasts. Each IP station is configured with the ATM address of the ARP server. IP stations set up SVCs with the ARP server, which in turn, sends InARP requests to the IP stations. Based on InARP reply, an ARP server sets up IP to ATM address maps. IP stations send ARP packets to the ARP server to resolve addresses, which returns ATM addresses. IP stations then set up a SVC to the destination station and data transfer begins. The ARP entries in IP stations and the ARP server age based on a well defined mechanism. For both the PVC and SVC environments, each IP station has at least one virtual circuit per destination address.

The Internet Engineering Task Force RFC2225 adds the support of ATM ARP Request Address list to RFC1577. The ATM ARP Request Address list is a list containing one or more ATM addresses of individual ATM ARP servers located within the LIS. The RFC2225 client eliminates the single point of failure associated with the 1577 clients' ATM ARP services. The 2225 clients have the ability to switch to backup ARP servers when the current ATM ARP server fails.

The client will always try to use the Primary ATM ARP server. If the effort to connect to the Primary ATM ARP server fails, the client tries to connect to the first Secondary server (the position in the ATM ARP Request Address list determines the order of the Secondary ATM ARP server). If the connection to the first Secondary ATM ARP server fails, the client tries to contact the next Secondary ATM ARP server in the list. This process continues until the connection is successful.

If the connection to the Primary ATM ARP server fails, regardless of which Secondary ATM ARP server it is connected to or attempting to connect to, the client continues to retry the Primary ATM ARP server every 15 minutes. If it finally connects to the Primary ATM ARP server, then the connection to the current Secondary ATM ARP server is dropped.

The ATM ARP Request Address list is entered manually either through SMIT or by using the **ifconfig** command. The ATM ARP Request Address list cannot be configured with the Management Information Base (MIB).

RARP Protocol Format

RARP is a protocol by which a physical machine in a local area network can request to learn its IP address from a gateway server's Address Resolution Protocol table or cache. This is needed since the machine may not have permanently attached disk where it can store its IP address permanently. A network administrator creates a table in a local area network's gateway router that maps the physical machine (or Medium Access Control - MAC) addresses to corresponding Internet Protocol addresses. When a new machine is set up, its RARP client program requests from the RARP server on the router to be sent its IP address. Assuming that an entry has been set up in the router table, the RARP server will return the IP address to the machine which can store it for future use.

Detailed Mechanism

Both the machine that issues the request and the server that responds use physical network addresses during their brief communication. Usually, the requester does not know the physical address. So, the request is broadcasted to all the machines on the network. Now, the requester must identify istelf uniquely to the server. For this either CPU serial number or the machine's physical network address can be used. But using the physical address as a unique id has two advantages.

- These addresses are always available and do not have to be bound into bootstrap code.
- Because the identifying information depends on the network and not on the CPU vendor, all machines on a given network will supply unique identifiers.

Request:

Like an ARP message, a RARP message is sent from one machine to the another encapsulated in the data portion of a network frame. An ethernet frame carrying a RARP request has the usual preamble, Ethernet source and destination addresses, and packet type fields in front of the frame.

The frame contains the value 8035 (base 16) to identify the contents of the frame as a RARP message. The data portion of the frame contains the 28-octet RARP message. The sender broadcasts a RARP request that specifies itself as both the sender and target machine, and supplies its physical network address in the target hardware address field. All machines on the network receive the request, but only those authorized to supply the RARP services process the request and send a reply, such machines are known informally as RARP servers. For RARP to succeed, the network must contain at least one RARP server.

Reply:

Servers answers request by filling in the target protocol address field, changing the message type from request to reply, and sending the reply back directly to the machine making the request.

Timing RARP Transactions

Since RARP uses the physical network directly, no other protocol software will time the response or retransmit the request. RARP software must handle these tasks. Some workstations that rely on RARP to boot, choose to retry indefinitely until the receive a response. Other implementations announce failure after only a few tries to avoid flooding the network with unnecessary broadcast.

Mulitple RARP Servers

Advantage: More reliability.

Disadvantage: Overloading may result when all servers respond. So, to get away with disadvantage we have primary and secondary servers. Each machine that makes RARP request is assigned a primary server. Normally, the primary server responds but if it fails, then requester may time out and rebroadcast the request. Whenever a secondary server receives a second copy of the request within a short time of the first, it responds. But, still there might be a problem that all secondary servers respond, thus overloading the network. So, the solution adopted is to avoid having all secondary servers transmit responses simultaneously. Each secondary server that receives the request computes a random delay and then sends a response.

Drawbacks of RARP

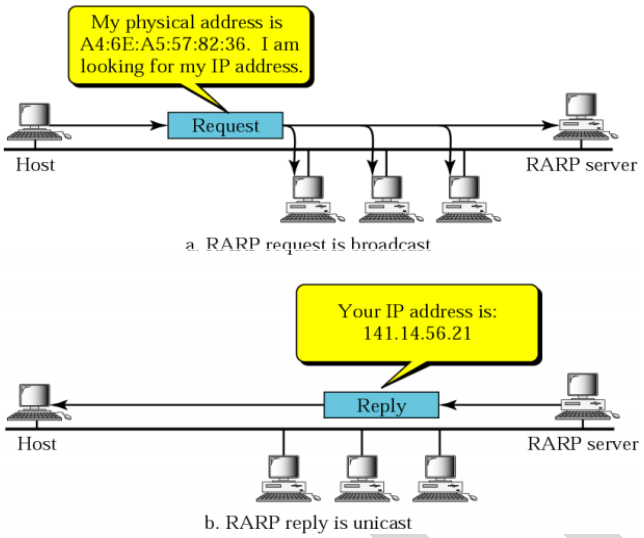
- Since it operates at low level, it requires direct addresss to the network which makes it difficult for an application programmer to build a server.
- It doesn't fully utilizes the capability of a network like ethernet which is enforced to send a minimum packet size since the reply from the server contains only one small piece of information, the 32-bit internet address.

Mapping physical address to logical address

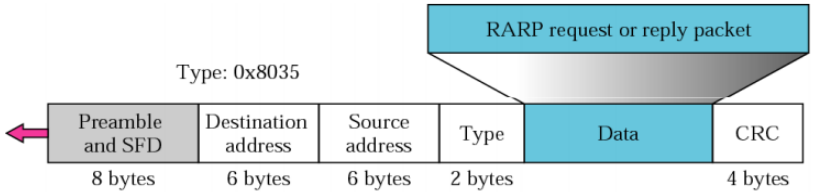
- In many situations, a host or router knows its MAC address but needs to know its logical (IP) addresses.
- **RARP** is used to find the logical addresses for a machine that knows its physical address.
- Each host or router can be assigned with one or more logical (IP) address/es. These addresses are unique and independent of the physical (hardware) address of the machine.
- To create an IP datagram, a host or router is required to know its own IP address. The IP address of a machine is generally read from its configuration file stored on a disk file. A diskless machine is booted from ROM, which has a minimum booting information.
- The machine gets its physical address (by reading NIC), which is unique locally. This physical address is used to get logical address by using RARP protocol.
- **RARP** request is created and broadcasted on the local network.
- So, the another machine on the local network that knows all the IP addresses responds with a RARP reply.
- It is necessary that the RARP requesting machine must be running a RARP client program and the responding machine must be running a RARP server program.

Problems with RARP

- Since it operates at low level, it requires direct address to the network, which makes it difficult for an application developer to built a server.
- It does not fully utilize the capability of a network, like ethernet, which is enforced to send a minimum packet size, since the reply from the server contains only small piece of information i.e. 32- bit internet address.



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)		

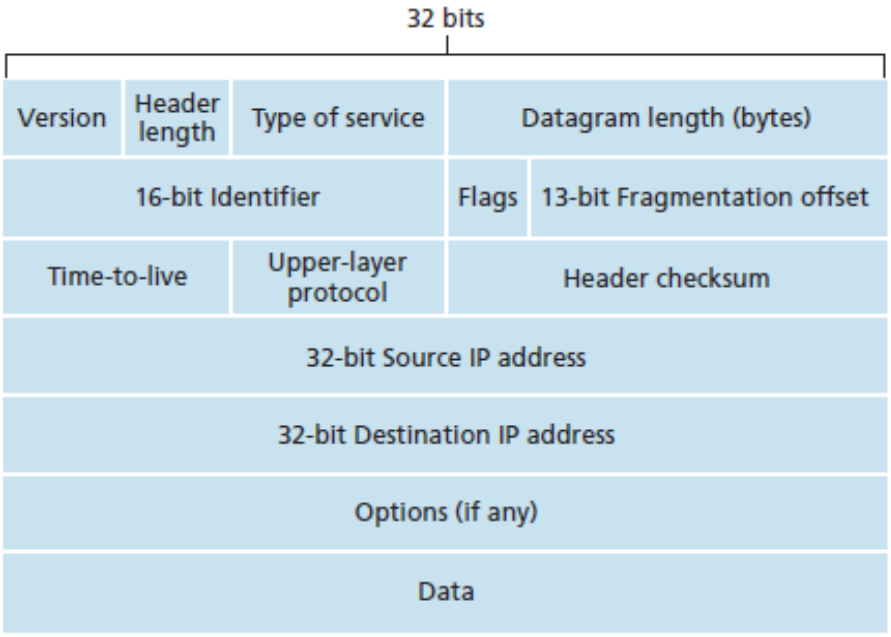


IP Datagram General Format

Data transmitted over an internet using IP is carried in messages called *IP datagrams*. Like all network protocol messages, IP uses a specific format for its datagrams. The IPv4 datagram is conceptually divided into two pieces: the *header* and the *payload*. The header contains addressing and control fields, while the payload carries the actual data to be sent over the internetwork. Unlike some message formats, IP datagrams do not have a footer following the payload.

Even though IP is a relatively simple, connectionless, "unreliable" protocol, the IPv4 header carries a fair bit of information, which makes it rather large. At a minimum, it is 20 bytes long, and with options can be significantly longer. The IP datagram format is described in Table 56 and illustrated in Figure 86.

The IPv4 datagram format is shown in the figure below.



Version Number

These 4 bits specify the IP protocol version of the datagram. By looking at the version number, the router can determine how to interpret the remainder of the IP datagram. Different versions of IP use different datagram formats. The datagram format for the current version of IP, IPv4 is shown in the figure above. The datagram format for the new version of IP (IPv6) will be discussed later.

Header Length

Because an IPv4 datagram can contain a variable number of options (which are included in the IPv4 datagram header), these 4 bits are needed to determine where in the IP datagram the data actually begins. Most IP datagrams do not contain options, so the typical IP datagram has a 20-byte header.

Type of Service

The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams (for example, datagrams particularly requiring low delay, high throughput, or reliability) to be distinguished from each other. For example, it might be useful to distinguish

real-time datagrams (such as those used by an IP telephony application) from non-real-time traffic (for example, FTP). The specific level of service to be provided is a policy issue determined by the router's administrator.

Datagram Length

This is the total length of the IP datagram (header plus data), measured in bytes. Since this filed is 16 bits long, the theoretical maximum size of the IP datagram is 65,535 bytes. However, datagrams are rarely larger than 1,500 bytes.

Identifier, Flags, Fragmentation Offset

These three fields have to do with so-called IP fragmentation, a topic we will consider in depth shortly. Interestingly, the new version of IP, IPv6, does not allow fragmentation at routers.

Time-to-live

The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever (due to, for example, a long-lived routing loop) in the network. This field is decremented by one each time the datagram is processed by a router. If the TTL field reaches 0, the datagram must be dropped.

Protocol

This field is used only when an IP datagram reaches its final destination. The value of this field indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed. For example, a value of 6 indicates that the data portion is passed to TCP, while a value of 17 indicates that the data is passed to UDP.

The 1s complement of this sum, known as the internet checksum, is stored in the checksum field. A router computes the header checksum of each received IP datagram and detects an error condition if the checksum carried in the datagram header does not equal the computed checksum. Routers typically discard datagrams for which an error has been detected.

Source and Destination IP Addresses

When a source creates a datagram, it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field. Often the source host determines the destination address via a DNS lookup.

Options

The options fields allow an IP header to be extended. Header options were meant to be used rarely – hence the decision to save overhead by not including the information in options fields in every datagram header.

Data (Payload)

Finally, we come to the last and most important field. In most circumstances, the data field of the IP datagram contains the transport-layer segment (TCP or UDP) to be delivered to the destination. However, the data field can carry other types of data, such as ICMP messages.

Note that an IP datagram has a total of 20 bytes of header (assuming no options). If the datagram carries a TCP segment, then each (nonfragmented) datagram carries a total of 40 bytes of header (20 bytes of IP header plus 20 bytes of TCP header) along with the application-layer message.

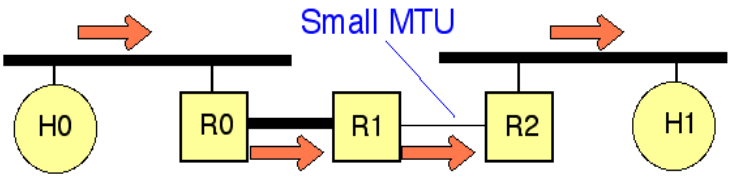
Example In an IP packet, the value of HLEN is 516 and the value of the total length field is 002816. How many bytes of data are being carried by this packet?

Solution The HLEN value is 5, which means the total number of bytes in the header is 5 × 4 or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

IP Fragmentation

An IP packet that is larger than the Maximum Transmission Unit (MTU) of an interface, is too large for transmission over that interface. The packet must either be fragmented, or discarded (and an ICMP error message returned to the sender). In either case, the original data will be

fragmented into smaller packets (less than the smallest MTU) in order to allow it to be received by the final destination system.

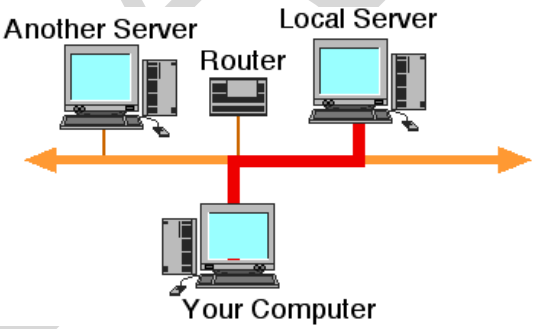


There are two approaches to doing this fragmentation:

- **IP Router Segmentation** - performing the fragmentation in the routers
- **IP Path MTU Discovery** - forcing the sender to perform the fragmentation

IP Fragmentation processing at a Router

The simplest approach from the end-system point of view is not to worry about the MTU size. In this simple approach, the sender simply has to ensure that each packet is less than the MTU of the link on which it is sent. (The router always knows this from the link interface configuration information).



Large IP packets that exceed the MTU of the link between R1 and R2 are fragmented by R1 in to two or more IP packets each smaller than the MTU size.

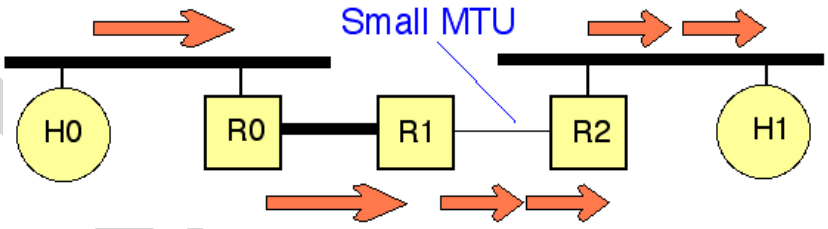
The network layer then has to arrange to cut packets up into smaller fragments whenever a router encounters a link with an MTU smaller than the received IP packet size. All the fragments of an IP packet carry the same ID in the IP packet header (allowing the final receiver to reassemble the

fragmented parts into the originalPDU). This is called "IP fragmentation" or "IP segmentation". The problem is, this offloads a lot of work on to routers, and in the worst case, can also result in packets being segmented by several IP routers one after another, resulting in very peculiar fragmentation.

Fragmentation Method

To fragment/segment a long internet packet, a router (R1 in the figure below) creates a new IP packet and copies the contents of the IP header fields from the long packet into the new IP header. The data of the long packet is then divided into two portions on a 8 byte (64 bit) boundary, so that the first packet is less than the MTU of the out-going interface. The more-fragments flag (MF) in the first packet is set to one (to indicate that more fragments of this packet follow). The More Flag may already be set in this packet if it has already been fragmented by another system. This packet is forwarded.

The second created new packet is then processed. The packet header field is identical to that of the original packet (including the same value of the packet ID, the total length field, the more-fragments flag (MF) and the fragment offset field in the original packet). The packet header field is updated with a new offset field, by adding the number of payload bytes sent in the first fragment. If this new packet is larger than the allowed link MTU, the packet is again fragmented.



IP Router Fragmentation

Any packet that has a more fragments (MF) flag set, must have an integral multiple of 8 bytes. (The final fragment, which does not have this flag set, may have an arbitrary number of bytes).

IP Router fragmentation is not recommended in the modern Internet, and this feature was not carried-forward when the next generation Internet Protocol (IPv6) was specified.

IP Fragmentation processing at a Sender

Path MTU Discovery allows a sender to fragment/segment a long internet packet, rather than relying on routers to perform IP-level fragmentation. This is more efficient and more scalable. It is therefore the recommended method in the current Internet. This is also the only method supported in IPv6.

IP Reassembly processing at the Receiving End System

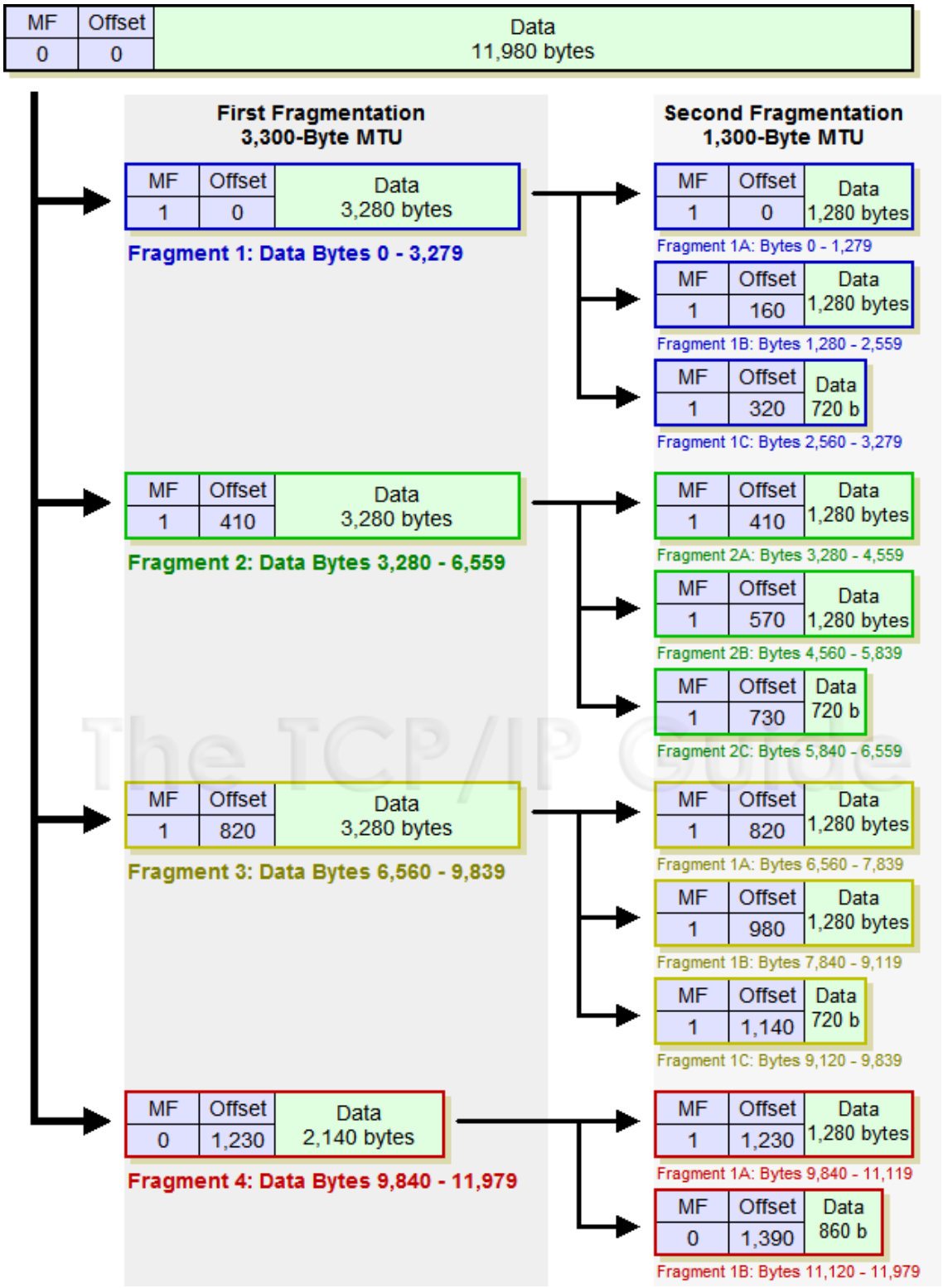
IP fragmentation and reassembly employs updating and using the values in the second 32 bits of the IPv4 packet header. An end system that accepts an IP packet (with a destination IP address that matches its own IP source address) will also reassemble any fragmented IP packets before these are passed to the next higher protocol layer.

The system stores all received fragments (i.e., IP packets with a more-fragments flag (MF) set to one, or where the fragment offset is non-zero), in one of a number of buffers (memory space). Packets with the same 16-bit Identification value are stored in the same buffer, at the offset specified by the fragment offset field specified in the packet header.

Packets which are incomplete remain stored in the buffer until either all fragments are received, OR a timer expires, indicating that the receiver does not expect to receive any more fragments. Completed packets are forwarded to the next higher protocol layer.

The IP Fragmentation Process: An Example

The device performing the fragmentation follows a specific algorithm to divide the message into fragments for transmission. The exact implementation of the fragmentation process depends on the device. Let's take the same example from the previous topic, an IP message 12,000 bytes wide (including the 20-byte IP header) that needs to be sent over a link with an MTU of 3,300. Here's a typical method by which this fragmentation might be performed .


MF 0 | Offset 0 | Data 11,980 bytes
First Fragmentation
3,300-Byte MTU
Second Fragmentation
1,300-Byte MTU
MF 1 | Offset 0 | Data 3,280 bytes
Fragment 1: Data Bytes 0 - 3,279
MF 1 | Offset 0 | Data 1,280 bytes
Fragment 1A: Bytes 0 - 1,279
MF 1 | Offset 160 | Data 1,280 bytes
Fragment 1B: Bytes 1,280 - 2,559
MF 1 | Offset 320 | Data 720 b
Fragment 1C: Bytes 2,560 - 3,279
MF 1 | Offset 410 | Data 3,280 bytes
Fragment 2: Data Bytes 3,280 - 6,559
MF 1 | Offset 410 | Data 1,280 bytes
Fragment 2A: Bytes 3,280 - 4,559
MF 1 | Offset 570 | Data 1,280 bytes
Fragment 2B: Bytes 4,560 - 5,839
MF 1 | Offset 730 | Data 720 b
Fragment 2C: Bytes 5,840 - 6,559
MF 1 | Offset 820 | Data 3,280 bytes
Fragment 3: Data Bytes 6,560 - 9,839
MF 1 | Offset 820 | Data 1,280 bytes
Fragment 1A: Bytes 6,560 - 7,839
MF 1 | Offset 980 | Data 1,280 bytes
Fragment 1B: Bytes 7,840 - 9,119
MF 1 | Offset 1,140 | Data 720 b
Fragment 1C: Bytes 9,120 - 9,839
MF 0 | Offset 1,230 | Data 2,140 bytes
Fragment 4: Data Bytes 9,840 - 11,979
MF 1 | Offset 1,230 | Data 1,280 bytes
Fragment 1A: Bytes 9,840 - 11,119
MF 0 | Offset 1,390 | Data 860 b
Fragment 1B: Bytes 11,120 - 11,979

IPv4 Datagram Fragmentation Process

In this diagram, the *MF* and *Fragment Offset* fields of each fragment are shown for reference. The *Data* fields are shown to scale (the length of each is proportional to the number of bytes in the fragment.)

1. **Create First Fragment:** The first fragment is created by taking the first 3,300 bytes of the 12,000-byte IP datagram. This includes the original header, which becomes the IP header of the first fragment (with certain fields changed as described below). So, 3,280 bytes of data are in the first fragment. This leaves 8,700 bytes to encapsulate (11,980 minus 3,280).
2. **Create Second Fragment:** The next 3,280 bytes of data are taken from the 8,700 bytes that remain after the first fragment was built, and paired with a new header to create fragment #2. This leaves 5,420 bytes.
3. **Create Third Fragment:** The third fragment is created from the next 3,280 bytes of data, with a 20-byte header. This leaves 2,140 bytes of data.
4. **Create Fourth Fragment:** The remaining 2,140 bytes are placed into the fourth fragment, with a 20-byte header of course.

In order to make the IP protocol independent of the physical network, the designers decided to make the maximum length of the IP datagram equal to 65,535 bytes. This makes transmission more efficient if we use a protocol with an MTU of this size. However, for other physical networks, we must divide the datagram to make it possible to pass through these networks. This is called fragmentation. The source usually does not fragment the IP packet. The transport layer will instead segment the data into a size that can be accommodated by IP and the data link layer in use. When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but some changed.

A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU. In other words, a datagram can be fragmented several times before it reaches the final destination. A datagram can be fragmented by the source host or any router in the path. There assembly of the datagram, however, is done only by the destination host because each fragment

becomes an independent datagram. An even stronger objection for reassembling packets during the transmission is the loss of efficiency it incurs.

When a datagram is fragmented, required parts of the header must be copied by all fragments.

The host or router that fragments a datagram must change the values of three fields: flags, fragmentation offset, and total length. The rest of the fields must be copied.

Fields Related to Fragmentation:

The fields that are related to fragmentation and reassembly of an IP datagram are the identification, flags, and fragmentation offset fields.

Identification:

i. This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IP address must uniquely define a datagram as it leaves the source host.

ii. To guarantee uniqueness, the IP protocol uses a counter to label the datagrams. The counter is initialized to a positive number.

iii. When the IP protocol sends a datagram, it copies the current value of the counter to the identification field and increments the counter by one.

iv. As long as the counter is kept in the main memory, uniqueness is guaranteed. When a datagram is fragmented, the value in the identification field is copied into all fragments.

v. In other words, all fragments have the same identification number, which is also the same as the original datagram.

vi. The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value should be assembled into one datagram.

Flags:

i. This is a three-bit field. The first bit is reserved (not used). The second bit is called the do not fragment bit.

ii. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary.

iii. The third bit is called the more fragment bit.

iv. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

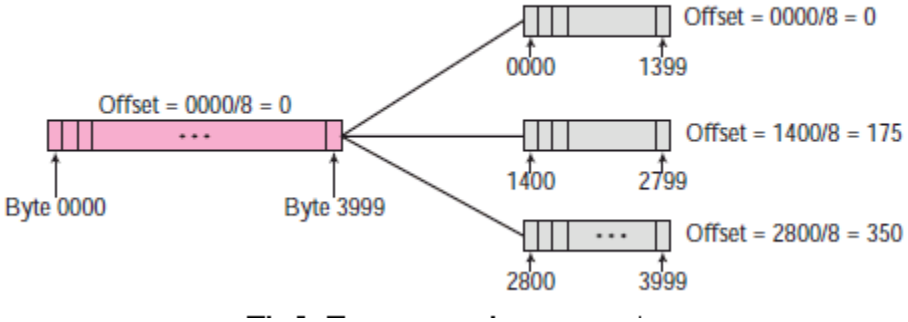
Fragmentation offset:

i. This 13-bit field shows the relative position of this fragment with respect to the whole datagram.

ii. It is the offset of the data in the original datagram measured in units of 8 bytes. Figure shows a datagram with a data size of 4000 bytes fragmented into three fragments. The bytes in the original datagram are numbered 0 to 3999. The first fragment carries bytes 0 to 1399.

iii. The offset for this datagram is 0/8= 0. The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8= 175.

iv. Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8= 350.



The value of the offset is measured in units of 8 bytes. This is done because the length of the offset field is only 13 bits long and cannot represent a sequence of bytes greater than 8191. This forces hosts or routers that fragment datagrams to choose the size of each fragment so that the first byte number is divisible by 8.

IP Datagram Options and Option Format

All IP datagrams must include the standard 20-byte header, which contains key information such as the source and destination address of the datagram, fragmentation control parameters, length information and more. In addition to these invariable fields, the creators of IPv4 included the ability to add *options* that provide additional flexibility in how IP handles datagrams. Use of these options is, of course, optional. However, all devices that handle IP datagrams must be capable of properly reading and handling them.

The IP datagram may contain zero, one or more options, which makes the total length of the *Options* field in the IP header variable. Each of the options can be either a single byte long, or multiple bytes in length, depending on how much information the option needs to convey. When more than one option is included they are just concatenated together and put into the *Options* field as a whole. Since the IP header must be a multiple of 32 bits, a *Padding* field is included if the number of bits in all options together is not a multiple of 32 bits.

IP Option Format

Each IP option has its own subfield format, generally structured as shown in Table. For most options, all three subfields are used: *Option Type*, *Option Length* and *Option Data*. For a few simple options, however, this complex substructure is not needed. In those cases, the option type itself communicates all the information required, so the *Option Type* field appears alone, while the *Option Length* and *Option Data* subfields are omitted.

Table :Internet Protocol Version 4 (IPv4) Option Format

<table>
<tr><th>Subfield Name</th><th>Size (bytes)</th><th>Description</th></tr>
<tr><td>Option Type</td><td>1</td><td>Option Type: This 8-bit field is divided into three "sub-subfields", according to the following format:
<table>
<tr><th>Sub-Subfield Name</th><th>Size (bytes)</th><th>Description</th></tr>
<tr><td>Copied</td><td>1/8
1 bit)</td><td>Copied Flag: This bit is set to 1 if the option is intended to be copied into all fragments when a datagram is fragmented; it is cleared to 0 if the option should not be copied into fragments.</td></tr>
<tr><td>Option Class</td><td>2/8
(2 bits)</td><td>Option Class: Specifies one of four potential values that indicate the general category into which the option belongs. In fact, only two of the values are used: 0 is for Control options, and 2 for Debugging and Measurement.</td></tr>
<tr><td>Option Number</td><td>5/8
(5 bits)</td><td>Option Number: Specifies the kind of option. 32 different values can be specified for each of the two option classes. Of these, a few are more commonly employed. See below for more information on the specific options.</td></tr>
</table></td></tr>
<tr><td>Option Length</td><td>0 or 1</td><td>Option Length: For variable-length options, indicates the size of the entire option, including all three subfields shown here, in bytes.</td></tr>
</table>

Option Data	0 or Variable	***Option Data:*** For variable-length options, contains data to be sent as part of the option.

IP Options

Table lists the most common IPv4 options, showing the option class, option number and length for each (a length of 1 indicating an option that consists of only an Option Type field), and providing a brief description of how each is used.

Table: Internet Protocol Version 4 (IPv4) Options

Option Class	Option Number	Length (bytes)	Option Name	Description
0	**0**	1	*End Of Options List*	An option containing just a single zero byte, used to mark the end of a list of options.
0	**1**	1	*No Operation*	A "dummy option" used as "internal padding" to align certain options on a 32-bit boundary when required.
0	**2**	11	*Security*	An option provided for the military to indicate the security classification of IP datagrams.
0	**3**	Variable	*Loose Source Route*	One of two options for source routing of IP datagrams. See below for an explanation.
0	**7**	Variable	*Record Route*	This option allows the route used by a datagram to be recorded within the header for the datagram itself. If a source device sends a datagram with

				this option in it, each router that "handles" the datagram adds its IP address to this option. The recipient can then extract the list of IP addresses to see the route taken by the datagram. Note that the length of this option is set by the originating device. It cannot be enlarged as the datagram is routed, and if it "fills up" before it arrives at its destination, only a partial route will be recorded.
0	**9**	Variable	*Strict Source Route*	One of two options for source routing of IP datagrams. See below for an explanation.
2	**4**	Variable	*Timestamp*	This option is similar to the *Record Route* option. However, instead of each device that handles the datagram inserting its IP address into the option, it puts in a timestamp, so the recipient can see how long it took for the datagram to travel between routers. As with the *Record Route* option, the length of this option is set by the originating device and cannot be enlarged by intermediate devices.
2	**18**	12	*Traceroute*	Used in the enhanced implementation of the traceroute utility, as described in RFC 1393. Also see the topic on the ICMP Traceroute messages.

IP Options and Source Routing

Normally, IP datagrams are routed without any specific instructions from devices regarding the path a datagram should take from the source to the destination. It's the job of routers, using routing protocols, to figure out those details. In some cases, however, it may be advantageous to have the source of a datagram specify the route a datagram takes through the network. This is called *source routing*.

There are two IP options that support source routing. In each, the option includes a list of IP addresses specifying the routers that must be used, to reach the destination. When *strict* source routing is used, this means that the path specified in the option must be used exactly, in sequence, with no other routers permitted to handle the datagram at all. In contrast, *loose* source routing specifies a list of IP addresses that must be followed in sequence, but having intervening hops in between the devices on the list is allowed.

Routing IP Datagrams

IP functions such as addressing, datagram encapsulation and if necessary, fragmentation and reassembly, all lead up to the ultimate objective of the protocol: the actual *delivery* of datagrams from a source device to one or more destination devices.

Unchanged Aspects of Datagram Delivery and Routing in IPv6

Most of the concepts related to how datagram delivery is accomplished in IPv6 are the same as in IPv4:

- Datagrams are delivered directly when the source and destination nodes are on the same network. When they are on different networks, delivery is indirect using routing to the destination's network, and then direct to the destination.
- Routing is performed by looking at IP addresses and determining which portion is the network ID and which the host ID. IPv6 does this in the same basic way as in classless IPv4, despite the fact that IPv6 unicast addresses are assigned using a special hierarchical format.
- Routing is still done on a next-hop basis, with sources generally not knowing how datagrams get from Point A to Point B.

- Routing is performed by devices called *routers* that maintain tables of routes that tell them where to forward datagrams to reach different destination networks.
- Routing protocols are used to allow routers to exchange information about routes and networks.

Changes in Datagram Delivery and Routing in IPv6

Most of the changes in routing in IPv6 are directly related to changes that we have seen in other areas of the protocol. Some of the main issues of note related to routing and routers in IPv6 include the following:

- **Hierarchical Routing and Aggregation:** One of the goals of the structure used for organizing unicast addresses was to improve routing. The unicast addressing format is designed to provide a better match between addresses and Internet topology, and to facilitate route aggregation. Classless addressing using CIDR in IPv4 was an improvement, but lacked any formal mechanism for creating a scalable hierarchy.
- **Scoped Local Addresses:** Local-use addresses including site-local and link-local are defined in IPv6, and routers must be able to recognize them. They must route them or *not* route them when appropriate. Multicast addresses also have various levels of scope.
- **Multicast and Anycast Routing:** Multicast is standard in IPv6, not optional as in IPv4, so routers must support it. Anycast addressing is a new type of addressing in IPv6.
- **More Support Functions:** Capabilities must be added to routers to support new features in IPv6. For example, routers play a key role in implementing serverless autoconfiguration and path MTU discovery in the new IPv6 fragmentation scheme.
- **New Routing Protocols:** Routing protocols such as RIP must be updated to support IPv6.
- **Transition Issues:** Last but certainly not least, routers play a major role in supporting the transition from IPv4 to IPv6. They will be responsible for connecting together IPv6 “islands” and performing translation to allow IPv4 and IPv6 devices to communicate with each other during the multi-year migration to the new protocol.

IP Routes and Routing Tables

Routers are responsible for forwarding traffic on an IP internetwork. Each router accepts datagrams from a variety of sources, examines the IP address of the destination and decides what the next hop is that the datagram needs to take to get it that much closer to its final destination. A question then naturally arises: how does a router know where to send different datagrams?

Each router maintains a set of information that provides a mapping between different network IDs and the other routers to which it is connected. This information is contained in a data structure normally called a *routing table*. Each entry in the table, unsurprisingly called a *routing entry*, provides information about one network (or subnetwork, or host). It basically says "if the destination of this datagram is in the following network, the next hop you should take is to the following device". Each time a datagram is received the router checks its destination IP address against the routing entries in its table to decide where to send the datagram, and then sends it on its next hop.

Obviously, the fewer the entries in this table, the faster the router can decide what to do with datagrams. Some routers only have connections to two other devices, so they don't have much of a decision to make. Typically, the router will simply take datagrams coming from one of its interfaces and if necessary, send them out on the other one. For example, consider a small company's router acting as the interface between a network of three hosts and the Internet. Any datagrams sent to the router from a host on this network will need to go over the router's connection to the router at the ISP.

When a router has connections to more than two devices, things become considerably more complex. Some distant networks may be more easily reachable if datagrams are sent using one of the routers than the other. The routing table contains information not only about the networks directly connected to the router, but also information that the router has "learned" about more distant networks.

Routing Tables in an Example Internetwork

Let's consider an example (see Figure) with routers R1, R2 and R3 connected in a "triangle", so that each router can send directly to the others, as well as to its own local network. Suppose R1's local network is 11.0.0.0/8, R2's is 12.0.0.0/8 and R3's is 13.0.0.0/8. R1 knows that any datagram it sees with 11 as the first octet is on its local network. It will also have a routing entry that says that any IP address starting with "12" should go to R2, and any starting with "13" should go to R3.

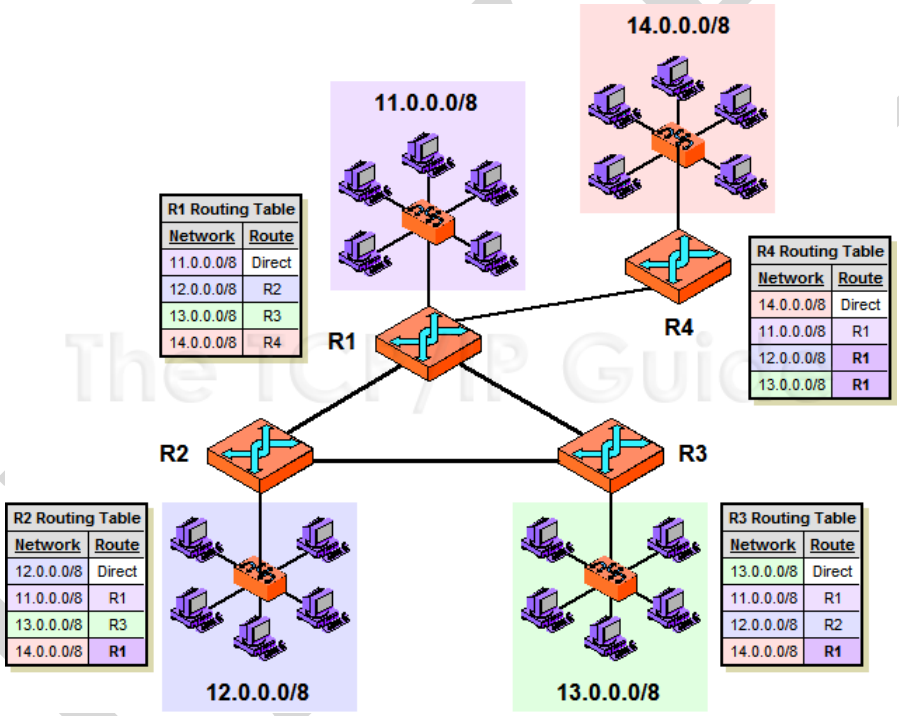


Figure IP Routing and Routing Tables

This diagram shows a small, simple internetwork consisting of four LANs each served by a router. The routing table for each lists the router to which datagrams for each destination network should be sent, and is color coded to match the colors of the networks. Notice that due to the "triangle", each of R1, R2 and R3 can send to each other. However, R2 and R3 must send through R1 to deliver to R4, and R4 must use R1 to reach either of the others.

Let's suppose that R1 also connects to another router, R4, which has 14.0.0.0/8 as its local network. R1 will have an entry for this local network. However, R2 and R3 also need to know how to reach 14.0.0.0/8, even though they don't connect to it its router directly. Most likely, they will have an entry that says that any datagrams intended for 14.0.0.0/8 should be sent to R1. R1 will then forward them to R4. Similarly, R4 will send any traffic intended for 12.0.0.0/8 or 13.0.0.0/8 through R1.

Route Determination

Now, imagine that this process is expanded to handle thousands of networks and routers. Not only do routers need to know which of their local connections to use for each network, they want to know, if possible, what is the ***best*** connection to use for each network. Since routers are interconnected in a mesh there are usually multiple routes between any two devices, but we want to take the best route whenever we can. This may be the shortest route, the least congested, or the route considered optimal based on other criteria.

Determining what routes we should use for different networks turns out to be an important but very complex job. Routers must plan routes and exchange information about routes and networks, which can be done in a variety of ways. This is accomplished in IP using special IP *routing protocols*. It is through these protocols that R2 and R3 would find out that 14.0.0.0/8 exists and that it is connected to them via R1.

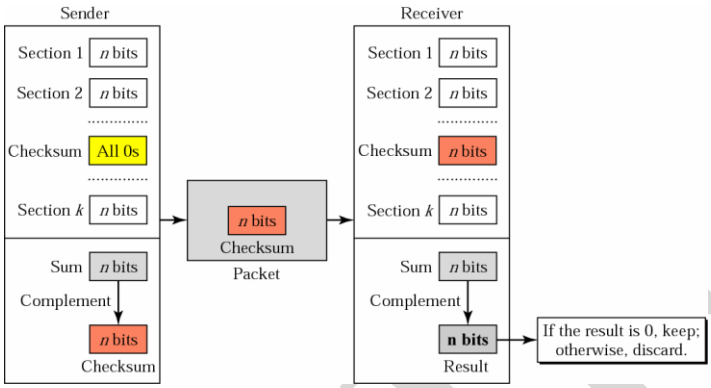
Checksum

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.

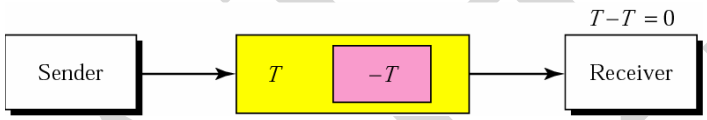
To create the checksum the sender does the following:

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

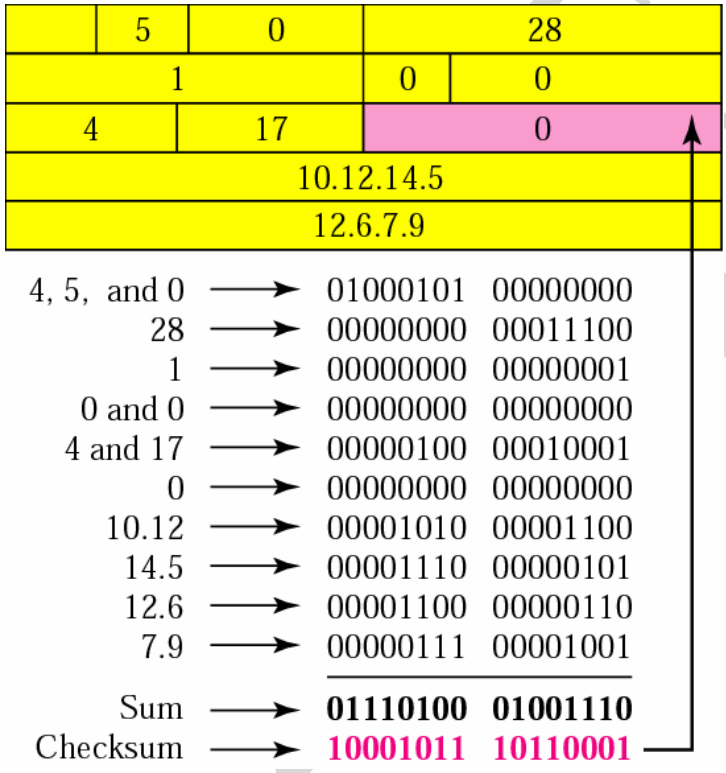


Checksum in the IP Packet

The implementation of the checksum in the IP packet follows the same principles discussed above. First, the value of the checksum field is set to 0. Then, the entire header is divided into 16-bit sections and added together. The result (sum) is complemented and inserted into the checksum field. The checksum in the IP packet covers only the header, not the data. There are two good reasons for this. First, all higher-level protocols that encapsulate data in the IP datagram have a checksum field that covers the whole packet. Therefore, the checksum for the IP datagram does not have to check the encapsulated data. Second, the header of the IP packet changes with each visited router, but the data do not. So the checksum includes only the part that has changed. If the data were included, each router would have to recalculate the checksum for the whole packet, which means an increase in processing time.

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.

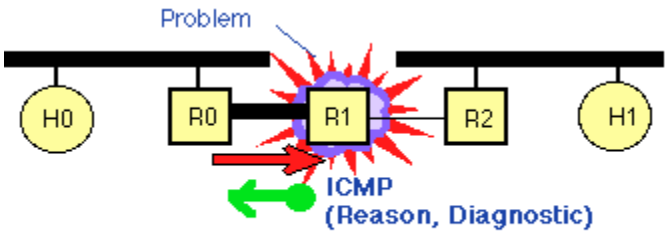


ICMP

ICMP (Internet Control Message Protocol) is an error-reporting protocol network devices like routers use to generate error messages to the source IP address when network problems prevent delivery of IP packets. ICMP creates and sends messages to the source IP address indicating that a gateway to the Internet that a router, service or host cannot be reached for packet delivery. Any IP network device has the capability to send, receive or process ICMP messages.

ICMP is *not* a transport protocol that sends data between systems. While ICMP is not used regularly in end-user applications, it is used by network administrators to troubleshoot Internet connections in diagnostic utilities including ping and traceroute. One of the main protocols of the Internet Protocolsuite, ICMP is used by routers, intermediary devices or hosts to communicate error information or updates to other routers, intermediary devices or hosts. The widely used IPv4 (Internet Protocol version 4) and the newer IPv6 use similar versions of the ICMP protocol

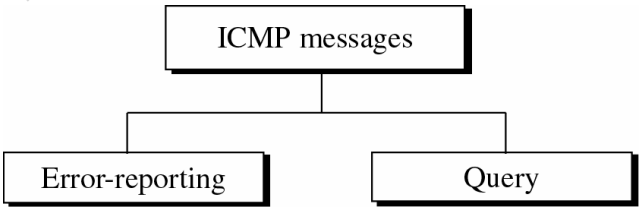
The protocol is also frequently used by Internet managers to verify correct operations of End Systems (ES) and to check that routers are correctly routing packets to the specified destination address.



Types of 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

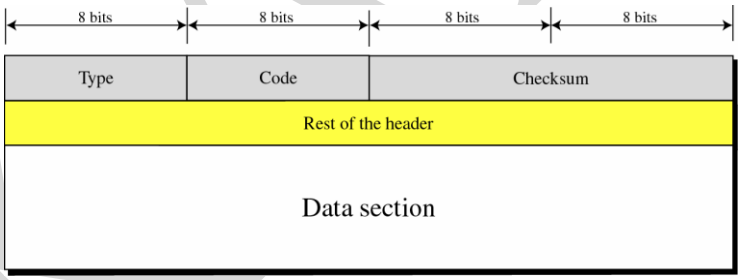
1) Error reporting
2) Query messages.



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.



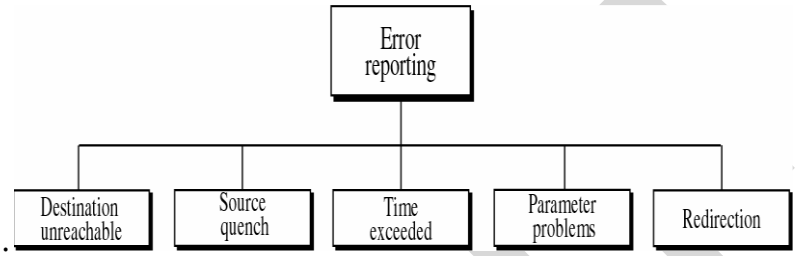
Type (8): specifies the type of ICMP message

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

Checksum (16): checksum of the entire ICMP message

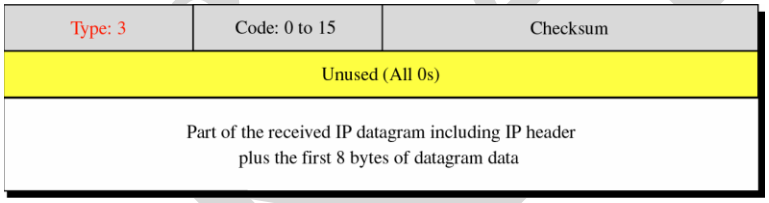
ERROR REPORTING

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

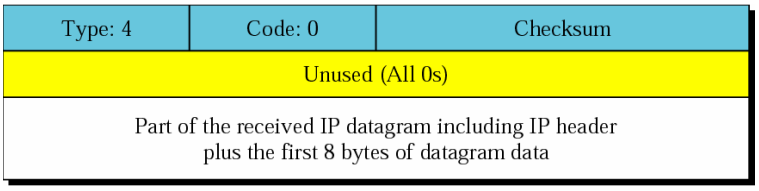
Code 13 Communication Administratively Prohibited

Code 14 Host Precedence Violation

Code 15 Precedence Cutoff Violation

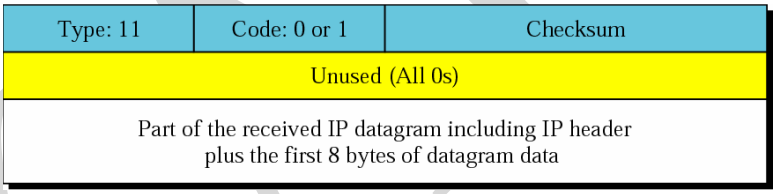
b) Source-quench

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



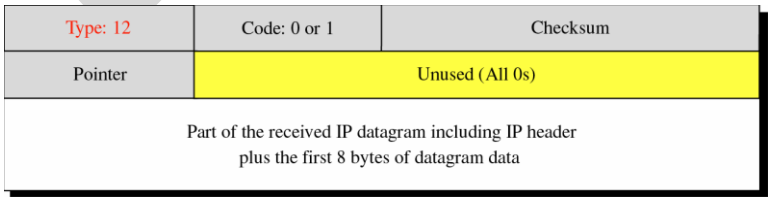
c) Time Exceed

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.

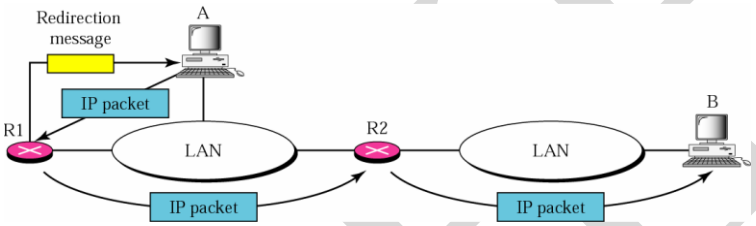


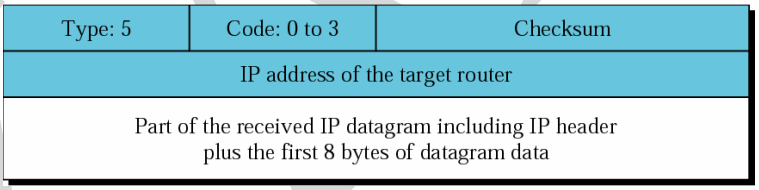
Code 0: Main header problem

Code 1: Problem in the option field

e) Redirection

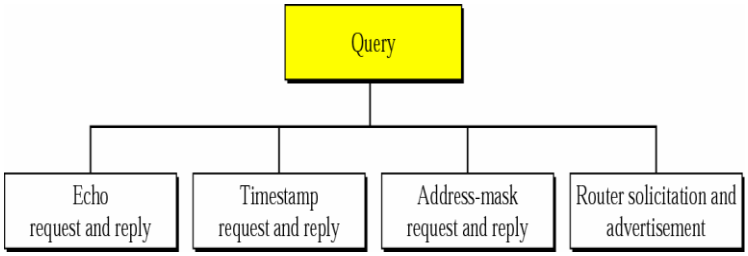
A host usually starts with a small routing table that is gradually augmented and updated. One of the tools to accomplish this is the redirection message. A redirection message is sent from a router to a host on the same local network.





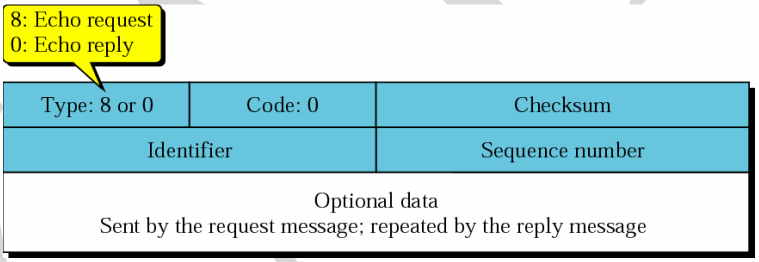
QUERY MESSAGES

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.



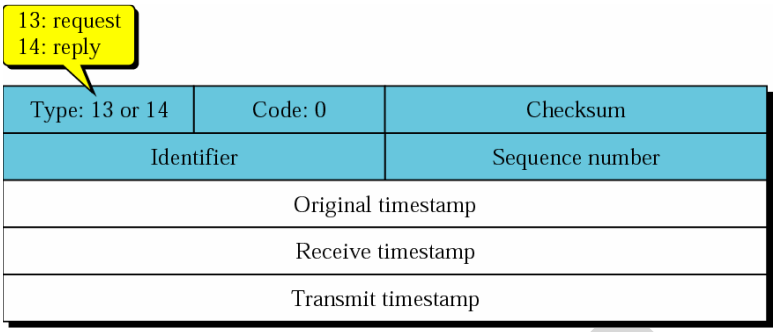
a) Echo Request and Reply:

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 reachability of a host. This is usually done by invoking the ping command.



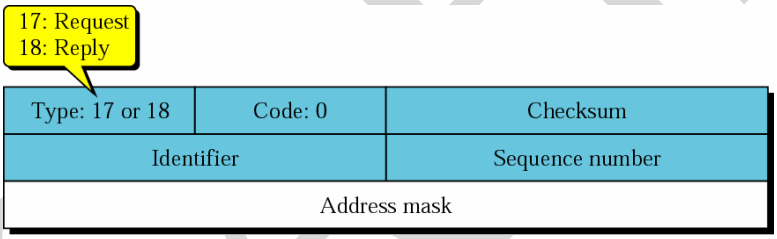
b) Timestamp Request and Reply

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.

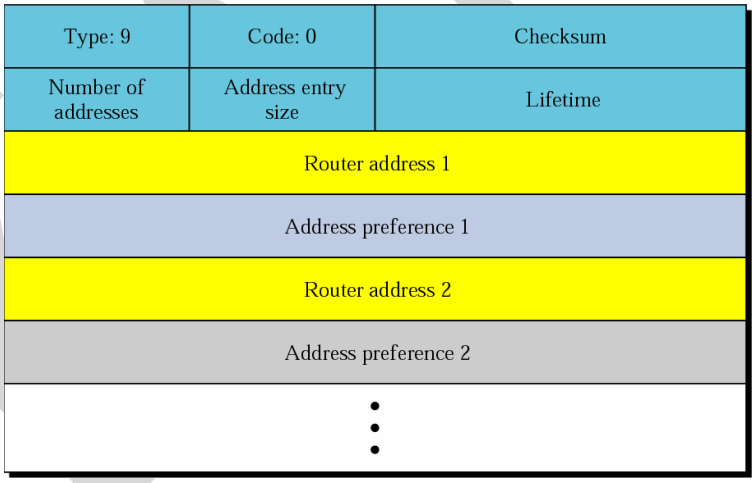


d) **Router-solicitation message**

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..



Checksum.

Checksum Calculation

The sender follows these steps using one's complement arithmetic:

1. The checksum field is set to zero.

2. The sum of all the 16-bit words (header and data) is calculated.

3. The sum is complemented to get the checksum.

4. The checksum is stored in the checksum field.

Checksum Testing

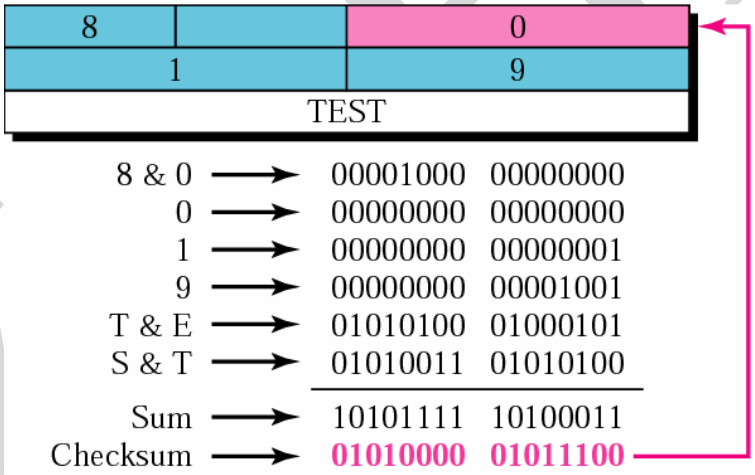
The receiver follows these steps using one's complement arithmetic:

1. The sum of all words (header and data) is calculated.

2. The sum is complemented.

3. If the result obtained in step 2 is 16 0s, the message is accepted; otherwise, it is rejected.

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.



POSSIBLE QUESTIONS

Two marks

1. What is ARP?
2. Mention the packet size of the RARP.
3. Mention the fields in ARP.
4. Define Proxy ARP.
5. Mention the usage of fragmentation.

6. List the fields in options.
7. List the usage of ICMP.
8. What are the types of messages in ICMP?
9. What is the necessity of checksum?
10. Draw the architecture of IP datagram.

Eight Marks

1. Brief about packet format and operation of ARP.
2. Give a brief description about the check sum in IP datagram.
3. Discuss about RARP operation.
4. Explain about ICMP protocol
5. Draw IP datagram format and give a brief description of each field in it in order.
6. Discuss about message format and error reporting in ICMP
7. Explain query message in ICMP.
8. Explain the process of fragmentation with an example.
9. Write about the option field in IP datagram.
10. Describe error reporting in ICMP.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

ONE MARK QUESTIONS

DEPARTMENT OF CS, CA & IT

STAFF NAME: S.MANJU PRIYA **SUBJECT NAME: INTERNETWORKING WITH TCP/IP**

SUB.CODE: 18CSP201 **UNIT II** **SEMESTER: II**

S.NO	Question	Choice1	Choice2	Choice3	Choice4	Ans
1	An ARP request is_____	Unicast	Broadcast	Telecast	mulitcast	Broadcast
2	An ARP reply is_____	Broadcast	Telecast	Unicast	mulitcast	Unicast
3	An ARP package involves_____components.	2	8	5	7	5
4	_____ Waits until an ARP Packet arrives.	Output module	Input module	Control module	super module	Input module
5	The error detection method used by most TCP/IP protocol is called the _____	checksum	parity	checkerror	error rectify	checksum
6	The maximum length of the diagram is _____ bytes	65,534	65,535	66,334	65,432	65,535

7	A datagram consist of _____ and _____	Title and data	Header and information	Content and header	Header and data	Header and data
8	ICMP stands for _____	Internet Control Message Protocol	Intranet Control Message	Internet Content Message	Intranet Content Message	Internet Control Message
9	ICMP is a _____ layer protocol	Application	Transport	Network	Physical	Network
10	ICMP messages are divided into _____ categories.	5	2	3	4	2
11	One of the main responsibilities of ICMP is to _____ errors.	Detect	Correct	Check	Report	Report
12	An ICMP message has _____ byte header.	8	6	4	2	8
13	There is no _____ mechanism in the IP protocol.	Data Control	Source Control	Flow Control	Error Control	Flow Control
14	A _____ message is send from router to a host on the same local network.	Redirection	Bidirection	Unidirection	Multidirection	Redirection
15	A _____ option is used to record the time of datagram processing.	Loose source Route	Time Stamp	Time Slicing	check sum	Time Stamp
16	An end of option is a _____ byte option.	4	6	8	1	1
17	MTU Stands for _____	Minimum Transfer Unit	Maximum Transfer Unit	Maximum Transport Unit	Minimum Transceiver Unit	Maximum Transfer Unit
18	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
19	The _____ contains the data that specific option requires.	Length field	data field	Width field	No field	data field

20	HLEN stands for _____	Header Length	Heading Length	Highlight Length Enable	Header Last	Header Length
21	_____ is a 3 bit sub field ranging from 0 to 7	Precedence	TOS Bits	Code point	protocol	Precedence
22	NIC stands for _____	Network Information Card	Network Information Center	Network Interface Card	Network Interface Center	Network Interface Card
23	_____ associates a logical address with physical address.	Static mapping	Dynamic mapping	Temporary mapping	Logical mapping	Static mapping
24	_____ addresses in the TCP/IP protocol suite are called IP address.	physical	static	dynamic	logical	logical
25	_____ defines the length of physical address in bytes.	protocol length	hardware length	software length	prototype length	hardware length
26	_____ define the length of logical address in bytes	protocol length	hardware length	software length	prototype length	protocol length
27	In cache table, _____ state means that the entry is complete.	Free	pending	resolved	cleared	resolved
28	TOS bits means	Type Of Service	Type Of Security	Type Of System	Type of Session	Type Of Service
29	_____ option used for padding at the end of the option field	end of option	checksum	operation option	no operation option	end of option
30	Destination unreachable message is _____ type of message.	Query	error reporting	query reporting	error detection	error reporting
31	address mask requesting or reply is _____ type of message.	error reporting	query	delay reporting	error detection	query
32	In destination _____represent the host is unreachable.	code1	code 2	code 3	code 4	code1

33	_____ command that can create series of echo request and echo reply.	ping	pong	pal	polar	ping
34	_____ program in units can be used to trace the route of a packet	Tracer	Trace Route	Trace up	Trace Down	Trace Route
35	_____ programs in windows can be used to trace the route of a packet	Tracer	Trace Route	Trace up	Trace Down	Tracer
36	_____ refers to finding network address	Multihome	Mask	Routing	subnet	Multihome
37	Vaiable length block is used in _____ address	Class address	classless addressing	Classful address	network addressing	classless addressing
38	The two terms often used in classless addressing is _____ and _____	address, type	length, prefix	Prefix, prefix length	suffix, suffix length.	Prefix, prefix length
39	In fixed length subnetting, the number of subnets is power of _____	4	5	3	2	5
40	Occasionally used term in classless addressing are _____ and _____	address, type	length, prefix	Prefix, prefix length	suffix, suffix length.	suffix, suffix length
41	All ARP request packets are transmitted with the _____	Ethernet broadcast address	Ethernet Unicast address	Ethernet Multicast address	Both a and c	Ethernet broadcast address
42	ARP reply packets is directed to the _____	Host	Router	Bridge	switch	Host
43	The size of an ARP request or reply packet is _____	24 Bytes	12 Bytes	6 Bytes	28 Bytes.	28 Bytes.
44	IP belongs to the _____ in the OSI model.	Network Layer	Session Layer	Transport Layer	Presentation Layer	Network Layer
45	ICMP Refers to _____	Internet Control Management Protocol	Internet Control Message	Internet Control Middleware	Internet Control Monitor Protocol	Internet Control Managemen

46	An _____ is a private access that uses the TCP/IP protocol suite	Internet	Extranet	Intranet	Ethernet	Intranet
47	A _____ network is totally isolated from the global Internet	Private	Hybrid	Virtual	LAN	Private

UNIT-III

Unicast Routing Protocol: Intra Domain and Inter Domain Routing – Distance Vector Routing – RIP – Link State Routing – OSPF – Path Vector Routing – BGP

Unicast Routing Protocol

Unicast routing is the process of forwarding unicasted traffic from a source to a destination on an internetwork. Unicasted traffic is destined for a unique address.

Intra Domain and Inter Domain Routing

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 intradomain routing protocols to handle routing inside the autonomous system. However, only one interdomain routing protocol handles routing between autonomous systems .

Distance Vector Routing

In distance vector routing, the least cost route between any two nodes is the route with minimum distance. In this protocol each node maintains a vector (table) of minimum distances to every node. The distance vector routing algorithm is sometimes called by other names, most commonly the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962). It was the original ARPANET routing algorithm and was also used in the Internet under the name RIP.

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar.

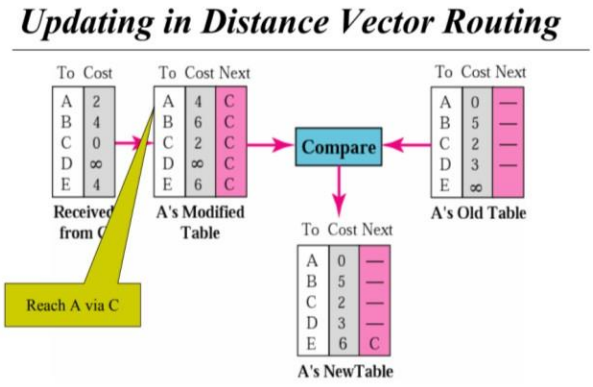
Initialization

At the beginning n Each node can know only the distance between itself and its immediate neighbors . We assume each node can send a message to the immediate neighbors and find the distance.

Sharing : Sharing of information between neighbors . In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

Updating

1. Receipt: a two-column table from a neighbor .
2. Add the cost between itself and the sending node to each value in the second column .
3. Repeat the following steps for each advertised destination
   - If (destination not in the routing table)
   - Add the advertised information to the table n Else
   - If (next-hop field is the same) n Replace retry in the table with the new advertised one
   - Else n If (advertised hop count smaller than one in the table)
   - Replace entry in the routing table



When to Share

1. The table is sent both periodically and when there is a change in the table
2. Periodic update
   - A node sends its routing table in a periodic update
   - Normally every 30 seconds
3. Triggered update
   - A node receives a table from a neighbor resulting in changes in its own table
   - A node detects some failure in the neighboring links which results in a distance change to infinity

Two-Node Loop Instability

A problem with distance vector routing is instability. A network using this protocol can become unstable

1. Both node A and B know how to reach node X

2. The link between A and X fails

   - Node A change its table

3a. if node A can send its routing table to B immediately

   - Everything is fine

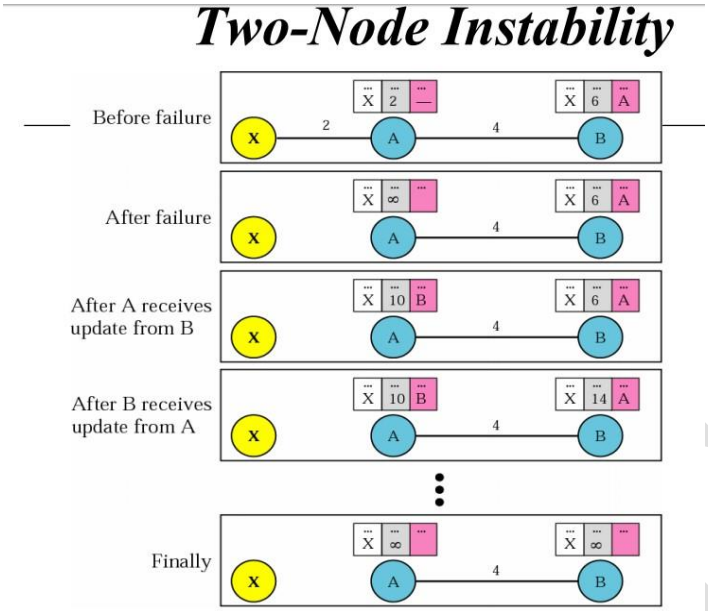
3b. However, if node B sends its routing table to A first

   - Node A assumes that B has found a way to reach X

4. A sends its new update to B and B also update its routing table

5. B sends its new update to A and so on…until the cost reach infinity

6. Then both A and B knows that the link is broken



As a result, during the time before cost reaches infinity

- A packet destined for X bounces between A and B
- Create a two-node loop problem

Solutions

- Defining infinity
- Split horizon
- Split horizon and poison reverse

Defining Infinity

Redefine infinity to a smaller number

- Shorten the time of instability

Most implementation define the distance between each node to be 1. Define 16 as infinity

As a result

- The distance vector scheme cannot be used in large system
- The size of network, in each direction, cannot exceed 15 hops

Split Horizon

Do not flood the table through each interface and a router must distinguish between different interface. If a router received route updating message from an interface. This same updated information must not be sent back through this interface. Since the information has come from the sending one.

Drawback of split horizon

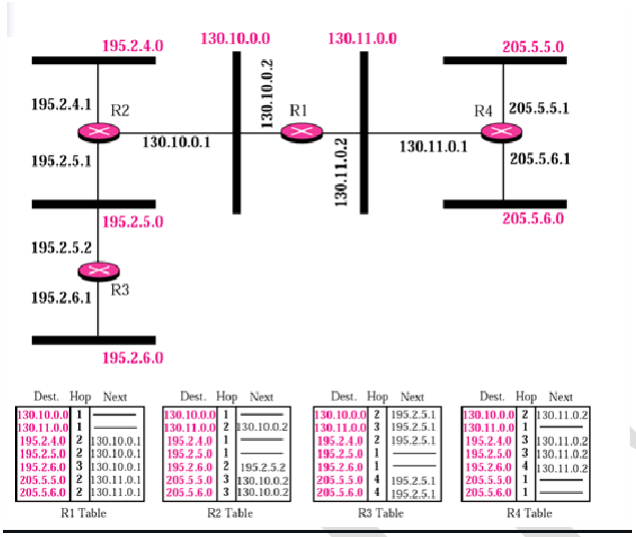
1. Distance vector uses a timer- If there is no news about a route within the time duration o Delete the route.
2. Since Node B eliminates the route to X . Node A cannot decide it is due to split horizon or because B has not received any news about X recently.

Poison Reverse

A variation of split horizons. Information received is used to update routing table and then passed out to all interface. However, a table entry is set to a metric of infinity as it' s come through and goes out interface are the same.

RIP

The Routing Information Protocol (RIP) is an intradomain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing. In a Autonomous System, RIP deals with routers and networks (links) n The destination in a routing table is a network . The first column defines a network address n The metric used in RIP is hop count n Infinity is defined as 16 .Any route in an AS cannot have more than 15 hops.



Example of a Domain Using RIP

RIP Message Format

1. Command: 8-bit

- The type of message: request (1) or response (2)

2. Version: 8-bit

- Define the RIP version

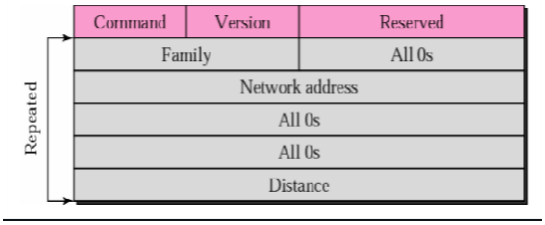
3. Family: 16-bit

- Define the family of the protocol used

- TCP/IP: value is 2

4. Network Address: 14 bytes .Defines the address of the destination network 14 bytes for this field to be applicable to any protocol .However, IP currently uses only 4 bytes, the rest are all 0s

5. Distance: 32-bit n. The hop count from the advertising router to the destination network.



RIP Message Format

Requests and Response

RIP uses two type of messages : Request and response .Request is sent by a router that has just come up or has some time-out entries . It can ask specific entries or all entries.

Response can be solicited or unsolicited . A solicited response: sent only in answer to a request It contains information about the destination specified in the corresponding request . An unsolicited response: sent periodically for every 30s. It contains information about the entire routing table . It is also called as update packet.

Timers in RIP

RIP uses three timers to support its operation (as shown in figure). The periodic timer controls the sending of messages, the expiration timer governs the validity of a route, and the garbage collection timer advertises the failure of a route.

Periodic Timer

The periodic timer controls the advertising of regular update messages. Although the protocol specifies that this timer must be set to 30 s, the working model uses a random number between 25 and 35 s. This is to prevent any possible synchronization and therefore overload on an internet if routers update simultaneously. Each router has one periodic timer that is randomly set to a number between 25 and 35. It counts down; when zero is reached, the update message is sent, and the timer is randomly set once again.

Expiration Timer

The expiration timer governs the validity of a route. When a router receives update information for a route, the expiration timer is set to 180 s for that particular route. Every time a new update for the route is received, the timer is reset. In normal situations this occurs every 30 s. However, if there is a problem on an internet and no update is received within the allotted 180 s, the route is considered expired and the hop count of the route is set to 16, which means the destination is unreachable. Every route has its own expiration timer.

Garbage Collection Timer

When the information about a route becomes invalid, the router does not immediately purge that route from its table. Instead, it continues to advertise the route with a metric value of 16. At the same time, a timer called the garbage collection timer is set to 120 s for that route. When the count reaches zero, the route is purged from the table. This timer allows neighbors to become aware of the invalidity of a route prior to purging.

Example : A routing table has 20 entries. It does not receive information about five routes for 200 s. How many timers are running at this time?

Solution

The 21 timers are listed below:

Periodic timer: 1

Expiration timer: $20 - 5 = 15$

Garbage collection timer: 5

RIP Version 2

RIP version 2 was designed to overcome some of the shortcomings of version 1. The designers of version 2 have not augmented the length of the message for each entry. They have only

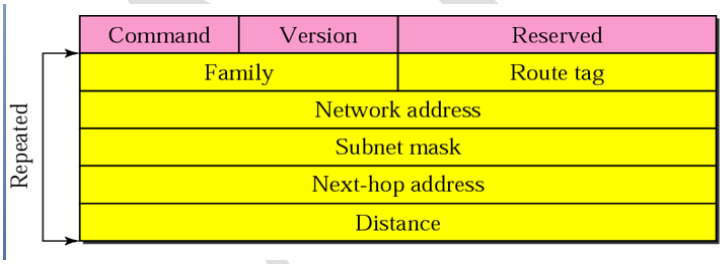
replaced those fields in version 1 that were filled with 0s for the TCP/IP protocol with some new fields. Message Format

The below figure shows the format of a RIP version 2 message. The new fields of this message are as follows:

1. Route tag. This field carries information such as the autonomous system number. It can be used to enable RIP to receive information from an interdomain routing protocol.

2. Subnet mask. This is a 4-byte field that carries the subnet mask (or prefix). This means that RIP2 supports classless addressing and CIDR.

3. Next-hop address. This field shows the address of the next hop. This is particularly useful if two autonomous systems share a network (a backbone, for example). Then the message can define the router, in the same autonomous system or another autonomous system, to which the packet next goes.

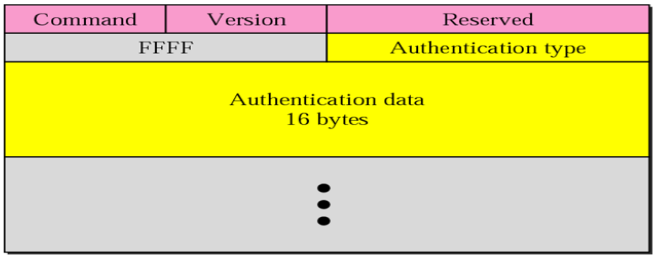


Classless Addressing

Probably the most important difference between the two versions of RIP is classful versus classless addressing. RIPv1 uses classful addressing. The only entry in the message format is the network address (with a default mask). RIPv2 adds one field for the subnet mask, which can be used to define a network prefix length. This means that in this version, we can use classless addressing. A group of networks can be combined into one prefix and advertised collectively.

Authentication

Authentication is added to protect the message against unauthorized advertisement. No new fields are added to the packet; instead, the first entry of the message is set aside for authentication information. To indicate that the entry is authentication information and not routing information, the value of FFFF16 is entered in the family field (as shown in figure). The second field, the authentication type, defines the protocol used for authentication, and the third field contains the actual authentication data.



Multicasting

Version 1 of RIP uses broadcasting to send RIP messages to every neighbor. In this way, all the routers on the network receive the packets, as well as the hosts. RIP version 2, on the other hand, uses the all-router multicast address to send the RIP messages only to RIP routers in the network.

Encapsulation

RIP messages are encapsulated in UDP user datagrams. A RIP message does not include a field that indicates the length of the message. This can be determined from the UDP packet. The well-known port assigned to RIP in UDP is port 520.

Link State Routing

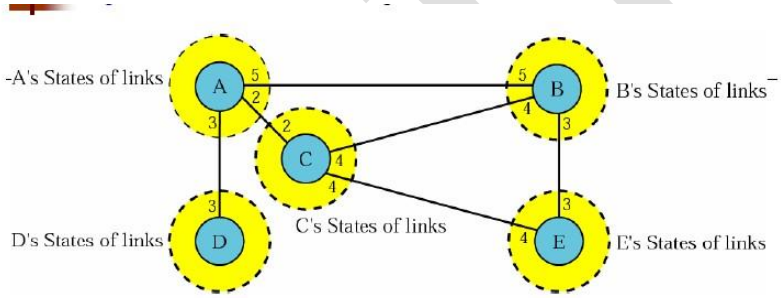
Link state routing has a different philosophy from that of distance vector routing. In link state routing, if each node in the domain has the entire topology of the domain— the list of nodes and

links, how they are connected including the type, cost (metric), and the condition of the links (up or down)—the node can use the Dijkstra algorithm to build a routing table.

Each node uses the same topology to create a routing table. But the routing table for each node is unique. Example: Like a city map

Assumption of link state routing

Although the global topology knowledge is not clear and each node has partial knowledge.It knows the state (type, condition, cost) of its link. However, the topology can be compiled from the partial knowledge of each node (as shown in below figure).



Link State Knowledge

Each node has a partial knowledge of the network. There is an overlap in the knowledge. The overlap guarantees the creation of a common topology.

Building Routing Tables

For sets of actions in link state routing creation of the states of the links by each node is formed which is called as link state packet or LSP. Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way. Formation of a shorten path tree for each node and Calculation of a routing table based on the shortest path tree is done.

Creation of Link State Packet (LSP)

A link state packet (LSP) can carry a large amount of information. For the moment, however, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age. The first two, node identity and the list of links, are needed to make

the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time.

LSPs are generated on two occasions:

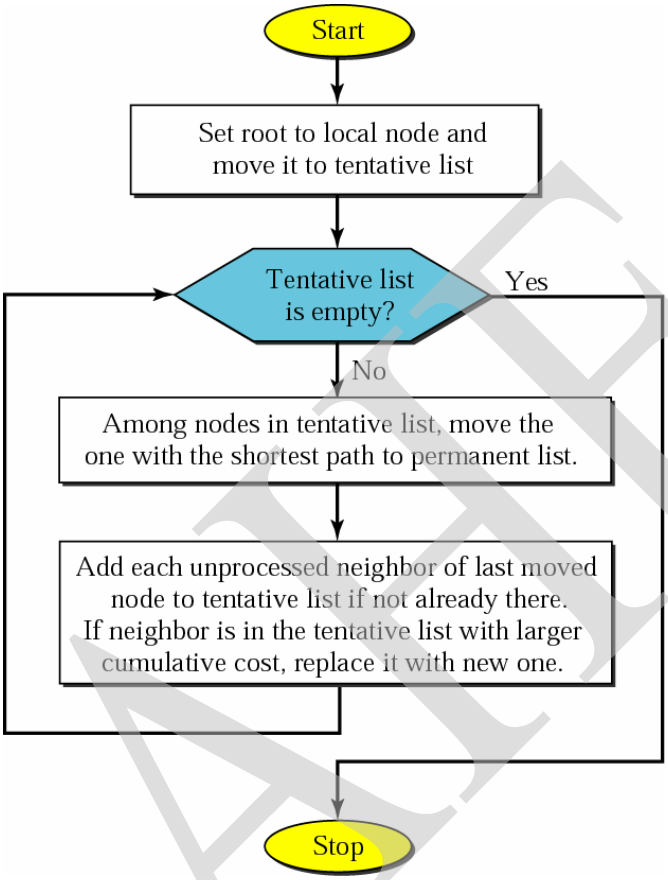
1. When there is a change in the topology of the domain. Triggering of LSP dissemination is the main way of quickly informing any node in the domain to update its topology.
2. On a periodic basis. The period in this case is much longer compared to distance vector routing. As a matter of fact, there is no actual need for this type of LSP dissemination. It is done to ensure that old information is removed from the domain. The timer set for periodic dissemination is normally in the range of 60 minutes or 2 hours based on the implementation. A longer period ensures that flooding does not create too much traffic on the network.

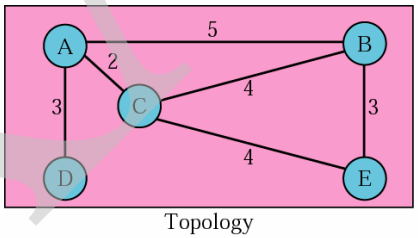
Flooding of LSPs

After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following:

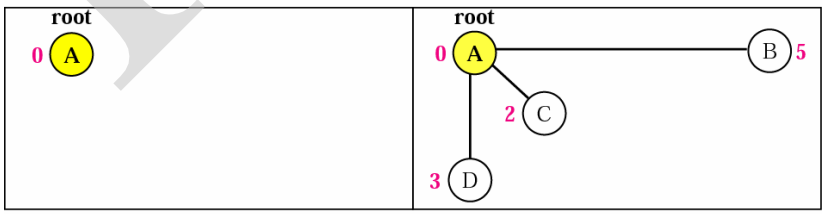
1. The creating node sends a copy of the LSP out of each interface.
2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following: a. It discards the old LSP and keeps the new one. b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

Formation of Shortest Path Tree: Dijkstra Algorithm



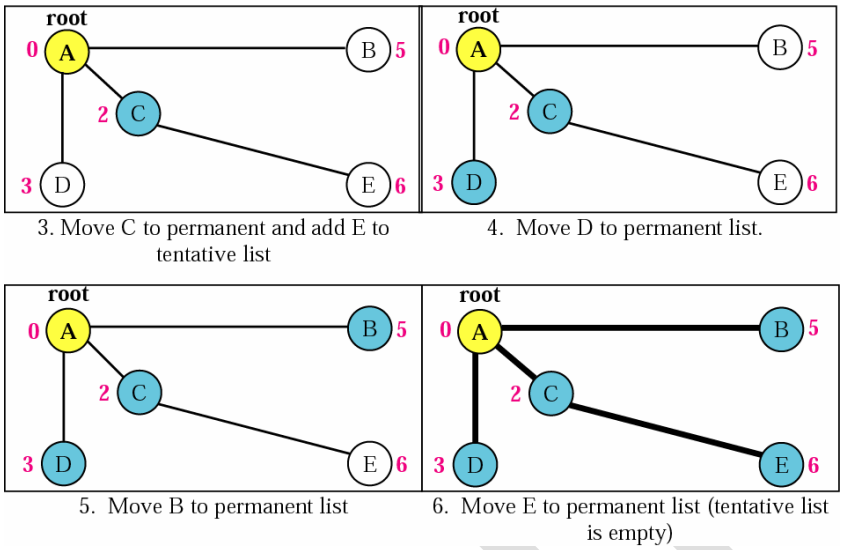


Topology



1. Set root to A and move A to tentative list

2. Move A to permanent list and add B, C, and D to tentative list



3. Move C to permanent and add E to tentative list

4. Move D to permanent list.

5. Move B to permanent list

6. Move E to permanent list (tentative list is empty)

Example of formation of shortest path tree

Calculation of Routing Table from Shortest Path Tree

Example:

Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

OSPF

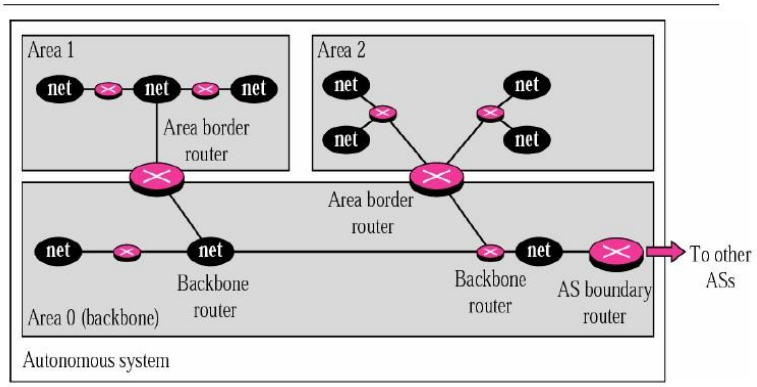
The Open Shortest Path First (OSPF) protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.

Areas

OSPF divides an autonomous system into areas to handle routing efficiently and in a timely manner.

1. A collection of networks, hosts, and routers all contained within an autonomous system
2. Thus, an autonomous system can be divided into many different areas
3. All networks inside an area must be connected

4. Routers inside an area flood the area with routing information
5. At the border of an area, special routers called area border routers
6. Summarize the information about the area and sent it to other areas
7. Among the area inside an autonomous system is a special area called backbone. All of the areas inside an AS must be connected to the backbone
8. The routers inside the backbone are called the backbone routers. A backbone router can also be an area border router
9. If the connectivity between a backbone and an area is broken, a virtual link must be created by the administration
10. Each area has area identification. The area identification of the backbone is zero



Areas in an autonomous systems

Metrics

OSPF allows the administrator to assign a cost, called the metric, to each route. Metric can be based on a type of service- Minimum delay, Maximum throughput. A router can have multiple routing tables. Each based on a different type of service.

Types of links

In OSPF, a connection is called a link. Four types of links

1. Point-to-point
2. Transient
3. Stub

4. Virtual

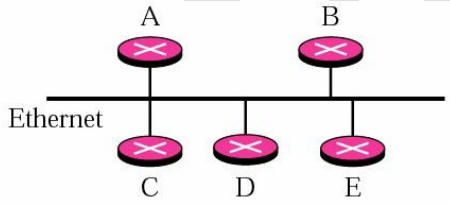
Point-to-point links

Connect two routers without any other host or router in these two routers.

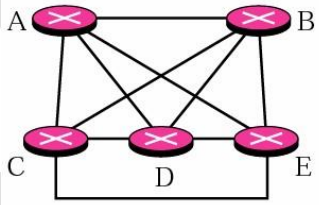
Example Telephone line**,** T-line

Transient Link

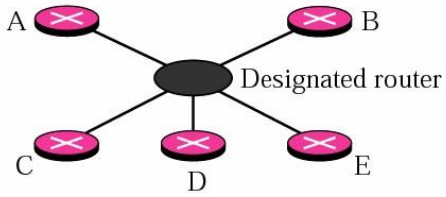
A transient link is a network with several routers attached to it. The data can enter through any of the routers and leave through any router. All LANs and some WANs with two or more routers are of this type. In this case, each router has many neighbors. For example, consider the Ethernet in shown in figure . Router A has routers B, C, D, and E as neighbors. Router B has routers A, C, D, and E as neighbors. If we want to show the neighborhood relationship in this situation, we have the graph as shown in figure.



a. Transient network



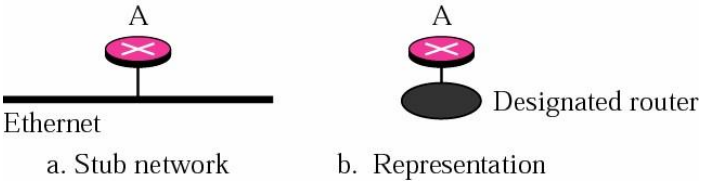
b. Unrealistic representation



c. Realistic representation

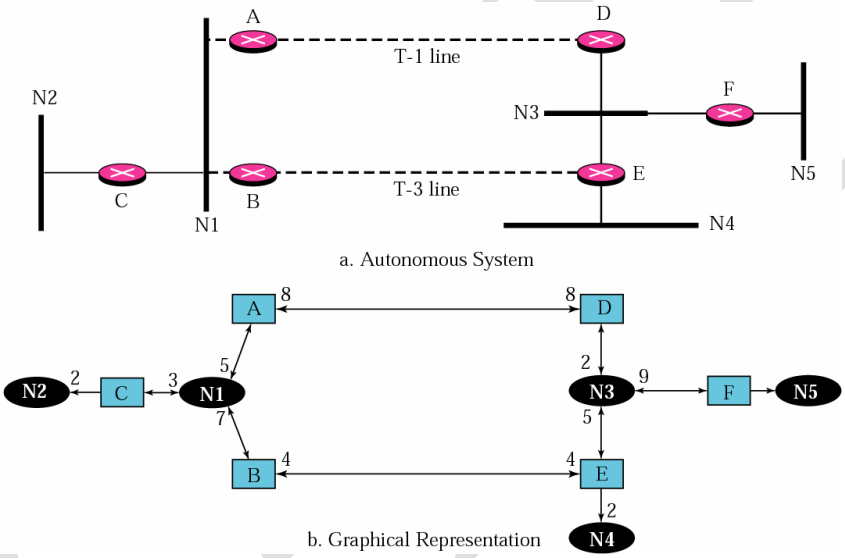
Stub Link

A stub link is a network that is connected to only one router. The data packets enter the network through this single router and leave the network through this same router. This is a special case of the transient network. We can show this situation using the router as a node and using the designated router for the network. However, the link is only onedirectional, from the router to the network (see Figure).



Virtual Link

When the link between two routers is broken, the administration may create a virtual link between them using a longer path that probably goes through several routers.



Example of an AS and its graphical representation in OSPF

OSPF Packets

OSPF uses five different types of packets: hello, database description, link state request, link state update, and link state acknowledgment. The most important one is the link state update that itself has five different kinds.

Common Header

All OSPF packets have the same common header (see Figure).

Version. This 8-bit field defines the version of the OSPF protocol. It is currently version .

Type. This 8-bit field defines the type of the packet. As we said before, we have five types, with values 1 to 5 defining the types.

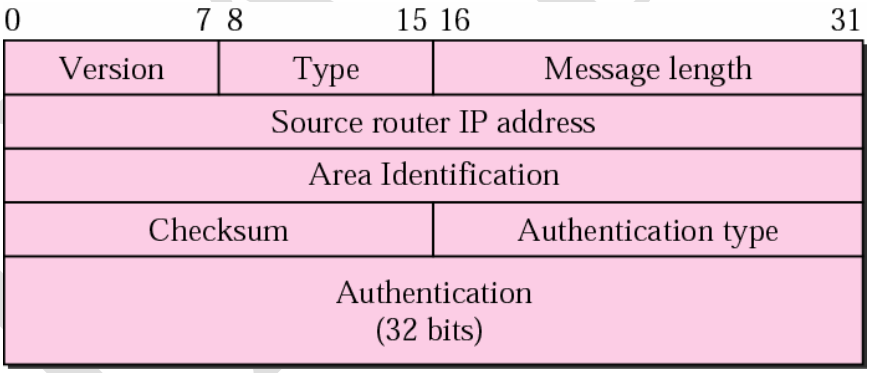
Message length. This 16-bit field defines the length of the total message including the header.

Source router IP address. This 32-bit field defines the IP address of the router that sends the packet. **Area identification**. This 32-bit field defines the area within which the routing takes place.

Checksum. This field is used for error detection on the entire packet excluding the authentication type and authentication data field.

Authentication type. This 16-bit field defines the authentication protocol used in this area. At this time, two types of authentication are defined: 0 for none and 1 for password.

Authentication. This 64-bit field is the actual value of the authentication data. In the future, when more authentication types are defined, this field will contain the result of the authentication calculation. For now, if the authentication type is 0, this field is filled with 0s. If the type is 1, this field carries an eight-character password.

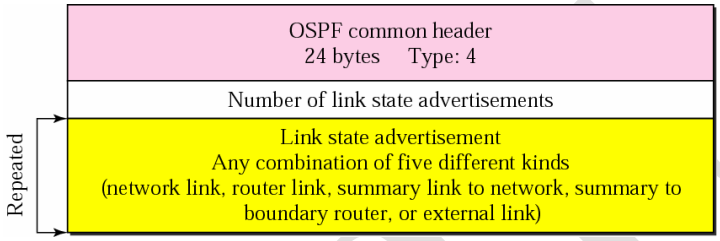


OSPF common header

Link State Update Packet

Used by a router to advertise the state of its Links. Each update packet may contain several different LSAs (List State Advertisement) Each update packet may contain several different LSAs. All five kinds have the same general header. This general header is shown in Figure and described below:

Link state age. This field indicates the number of seconds elapsed since this message was first generated. Recall that this type of message goes from router to router (flooding). When a router creates the message, the value of this field is 0. When each successive router forwards this message, it estimates the transit time and adds it to the cumulative value of this field.



Link state update packet

E flag. If this 1-bit flag is set to 1, it means that the area is a stub area. A stub area is an area that is connected to the backbone area by only one path.

T flag. If this 1-bit flag is set to 1, it means that the router can handle multiple types of service.

Link state type. This field defines the LSA type. As we discussed before, there are five different advertisement types: router link (1), network link (2), summary link to network (3), summary link to AS boundary router (4), and external link (5).

Link state ID. The value of this field depends on the type of link. For type 1 (router link), it is the IP address of the router. For type 2 (network link), it is the IP address of the designated router. For type 3 (summary link to network), it is the address of the network. For type 4 (summary link to AS boundary router), it is the IP address of the AS boundary router. For type 5 (external link), it is the address of the external network.

Advertising router. This is the IP address of the router advertising this message.

Link state sequence number. This is a sequence number assigned to each link state update message. **Link state checksum**. This is not the usual checksum. Instead, the value of this field is calculated using Fletcher's checksum ,which is based on the whole packet except for the age field.

Length. This defines the length of the whole packet in bytes.

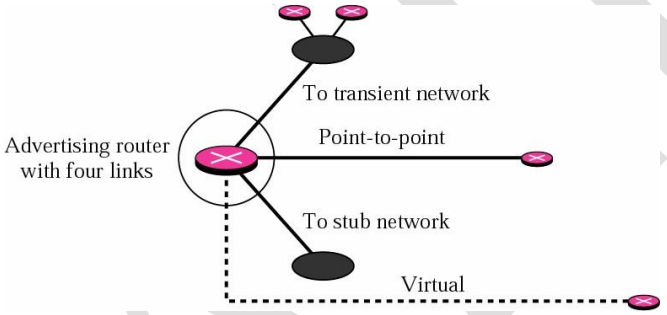
Router Link LSA

A router link defines the links of a true router. A true router uses this advertisement to announce information about all of its links and what is at the other side of the link (neighbors). See Figure for a depiction of a router link. The router link LSA advertises all of the links of a router (true router). The format of the router link packet is shown in Figure.

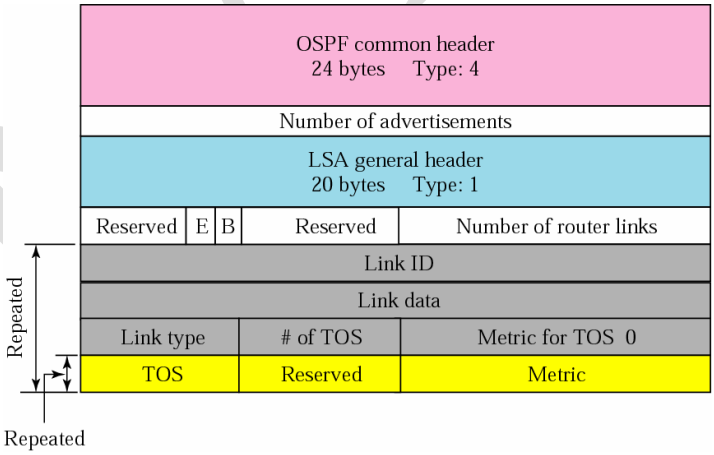
The fields of the router link LSA are as follows:

Link ID. The value of this field depends on the type of link. Table below shows the different link identifications based on link type.

Link data. This field gives additional information about the link. Again, the value depends on the type of the link (see Table)



Router link



Router Link LSA

Link type. Four different types of links are defined based on the type of network to which the router is connected (see Table).

Link Type	*Link Identification*	*Link Data*
Type 1: Point-to-point	Address of neighbor router	Interface number
Type 2: Transient	Address of designated router	Router address
Type 3: Stub	Network address	Network mask
Type 4: Virtual	Address of neighbor router	Router address

Link Types, Link Identification, and Link Data

Number of types of service (TOS). This field defines the number of types of services announced for each link.

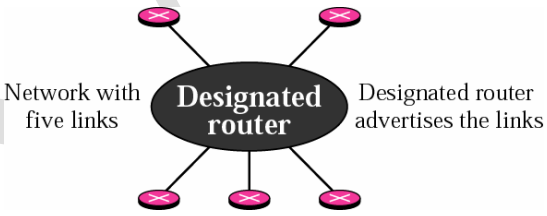
Metric for TOS 0. This field defines the metric for the default type of service (TOS 0).

TOS. This field defines the type of service.

Metric. This field defines the metric for the corresponding TOS

Network Link LSA

A network link defines the links of a network. A designated router, on behalf of the transient network, distributes this type of LSP packet. The packet announces the existence of all of the routers connected to the network (see Figure).



Network Link

The format of the network link advertisement is shown in Figure. The fields of the network link LSA are as follows:

Network mask. This field defines the network mask.

Attached router. This repeated field defines the IP addresses of all attached routers.

Summary Link to Network LSA

Router link and network link advertisement flood the area with information inside an area. But a router must also know about the networks outside its area. The *area border routers* provide this information. An area border router is active in more than one area.

Receive *router link* and *network link advertisements* create a router table for each area and p rovide one area's information to other areas by the *summary link to network advertisement*

Summary Link to AS Boundary Router LSA

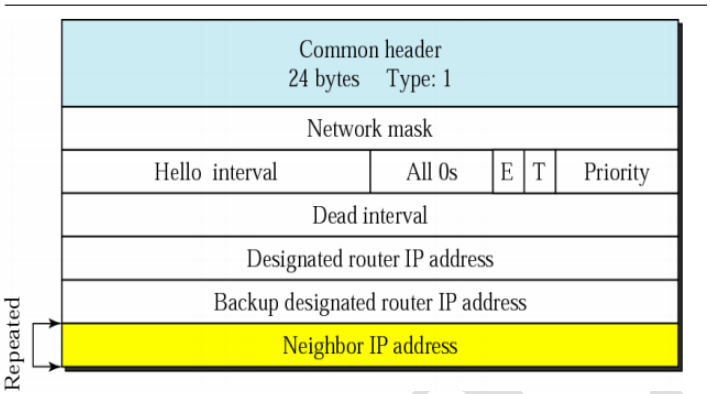
The LSA consists of only network mask and metric for each type of service.It not include the *network address.*Since the IP address of the advertising router is in the header each advertisement announces *only one network*. If more than one network, a separate advertisement must be issued for each.

External Link LSA

Although the previous advertisement lets each router know the route to an AS boundary router, this information is not enough. A router inside an autonomous system wants to know which networks are available outside the autonomous system; the external link advertisement provides this information. The AS boundary router floods the autonomous system with the cost of each network outside the autonomous system using a routing table created by an interdomain routing protocol. Each advertisement announces one single network. If there is more than one network, separate announcements are made.

Other Packets

Hello Message OSPF uses the hello message to create neighborhood relationships and to test the reachability of neighbors. This is the first step in link state routing. Before a router can flood all of the other routers with information about its neighbors, it must first greet its neighbors. It must know if they are alive, and it must know if they are reachable.



Network mask. This 32-bit field defines the network mask of the network over which the hello message is sent.

Hello interval. This 16-bit field defines the number of seconds between hello messages.

E flag. This is a 1-bit flag. When it is set, it means that the area is a stub area.

T flag. This is a 1-bit flag. When it is set, it means that the router supports multiple metrics.

Priority. This field defines the priority of the router. The priority determines the selection of the designated router. After all neighbors declare their priorities, the router with the highest priority is chosen as the designated router. The one with the second highest priority is chosen as the backup designated router. If the value of this field is 0, it means that the router never wants to be a designated or a backup designated router.

Dead interval. This 32-bit field defines the number of seconds that must pass before a router assumes that a neighbor is dead.

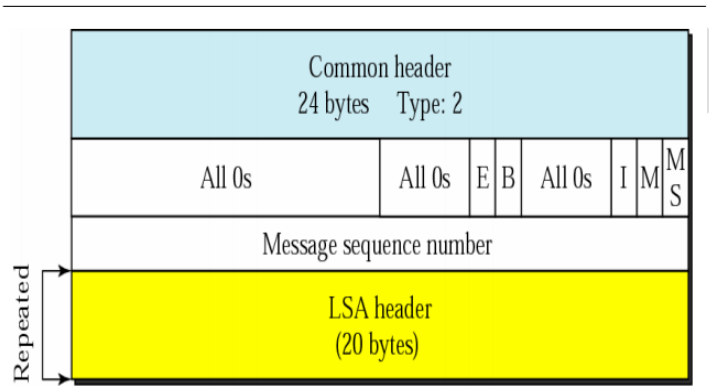
Designated router IP address. This 32-bit field is the IP address of the designated router for the network over which the message is sent.

Backup designated router IP address. This 32-bit field is the IP address of the backup designated router for the network over which the message is sent.

Neighbor IP address. This is a repeated 32-bit field that defines the routers that have agreed to be the neighbors of the sending router. In other words, it is a current list of all the neighbors from which the sending router has received the hello message.

Database Description Message

When a router is connected to the system *for the first time* or *after a failure.* It needs the complete link state database immediately. Thus, it sends hello packets to greet its neighbors. If this is the first time that the neighbors hear from the router. They send a *database description packet.* The database description message does not contain complete database information. It only gives an *outline*, the title of each line in the database. The newly router examines the outline and find out which lines it does not have. Send one or more *link state request packets* to get full information about that particular link. The content of the database may be divided into several message. When two routers want to exchange database description packets, One of them acts as mater and the other is the slave.



E flag. This 1-bit flag is set to 1 if the advertising router is an autonomous boundary router (E stands for external).

B flag. This 1-bit flag is set to 1 if the advertising router is an area border router.

I flag. This 1-bit field, the initialization flag, is set to 1 if the message is the first message.

M flag. This 1-bit field, the more flag, is set to 1 if this is not the last message.

M/S flag. This 1-bit field, the master/slave bit, indicates the origin of the packet: master (M/S = 1) or slave (M/S = 0).

Message sequence number. This 32-bit field contains the sequence number of the message. It is used to match a request with the response.

LSA header. This 20-byte field is used in each LSA. The format of this header is discussed in the link state update message section. This header gives the outline of each link, without details. It is repeated for each link in the link state database.

Link State Request Packet

The format of the link state request packet is shown in Figure. This is a packet that is sent by a router that needs information about a specific route or routes. It is answered with a link state update packet. It can be used by a newly connected router to request more information about some routes after receiving the database description packet. The three fields here are part of the LSA header, which has already been discussed. Each set of the three fields is a request for one single LSA. The set is repeated if more than one advertisement is desired.

Link State Acknowledgment Packet

OSPF makes routing more reliable by forcing every router to acknowledge the receipt of every link state update packet. The format of the link state acknowledgment packet is shown in Figure . It has the common OSPF header and the general LSA header. These two sections are sufficient to acknowledge a packet.

Encapsulation

OSPF packets are encapsulated in IP datagrams. They contain the acknowledgment mechanism for flow and error control. They do not need a transport layer protocol to provide these services.

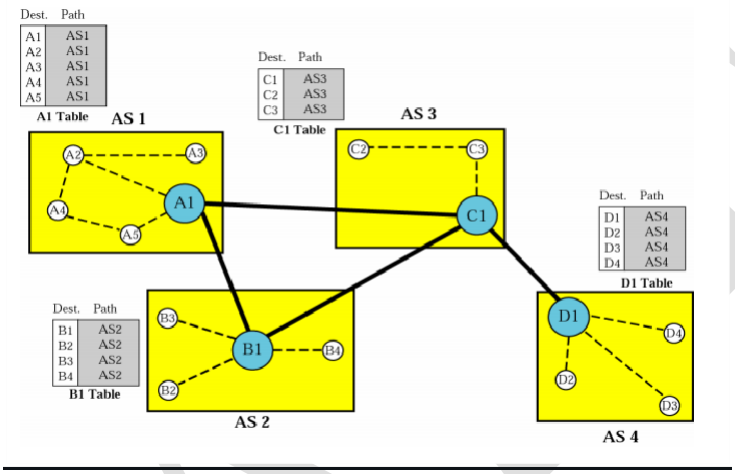
Path Vector Routing

Path vector routing is similar to distance vector routing. There is at least one node, called the speaker node, in each AS that creates a routing table and advertises it to speaker nodes in the neighboring ASs.. Path vector routing is exterior routing protocol proved to be useful for interdomain or inter-AS routing as it is sometimes called. In distance vector routing, a router has a list of networks that can be reached in the same AS with the corresponding cost (number of hops). In path vector routing, a router has a list of networks that can be reached with the path

(list of ASs to pass) to reach each one. In other words, the domain of operation of the distance vector routing is a single AS; the domain of operation of the path vector routing is the whole Internet. The distance vector routing tells us the distance to each network; the path vector routing tells us the path.

Initialization

At the beginning, each speaker node can know only the reachability of nodes inside its AS. Example is shown in figure.

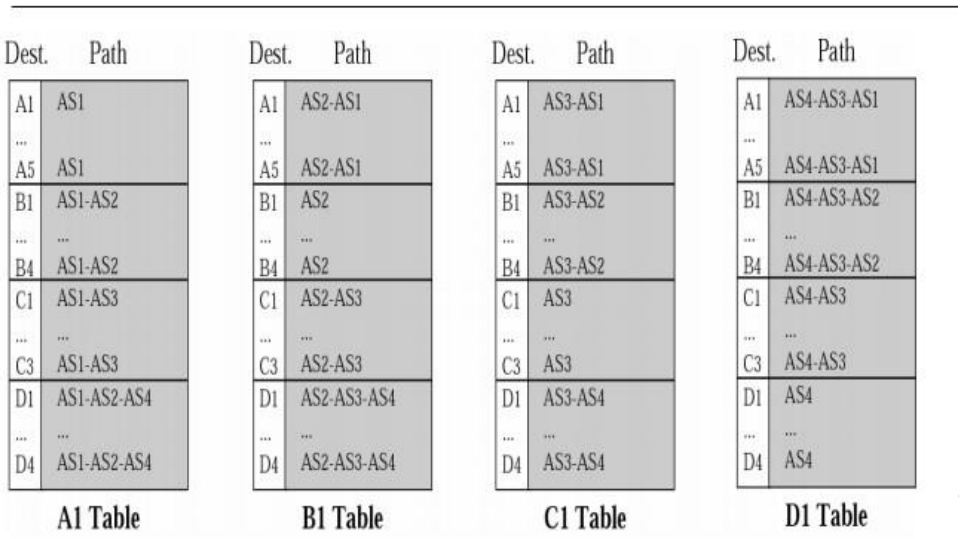


Sharing

A speaker in an AS share its table with immediate neighbors. Example : A1 share with B1 and C1.

Updating

When a speaker node receives a two-column tables, update its table. It adds the nodes that are not in its routing table. Add its own AS and AS that sent the table. Example: The figure shows a stabilized table. If A1 receives a packet for node A3. The path is in AS1 (the packet is at home). If D1 receives a packet for node A2. The packet should go from AS4 to AS3, and then to AS1.



Dest.	Path
A1	AS1
...	
A5	AS1
B1	AS1-AS2
...	...
B4	AS1-AS2
C1	AS1-AS3
...	...
C3	AS1-AS3
D1	AS1-AS2-AS4
...	...
D4	AS1-AS2-AS4

A1 Table

Dest.	Path
A1	AS2-AS1
...	
A5	AS2-AS1
B1	AS2
...	...
B4	AS2
C1	AS2-AS3
...	...
C3	AS2-AS3
D1	AS2-AS3-AS4
...	...
D4	AS2-AS3-AS4

B1 Table

Dest.	Path
A1	AS3-AS1
...	
A5	AS3-AS1
B1	AS3-AS2
...	...
B4	AS3-AS2
C1	AS3
...	...
C3	AS3
D1	AS3-AS4
...	...
D4	AS3-AS4

C1 Table

Dest.	Path
A1	AS4-AS3-AS1
...	
A5	AS4-AS3-AS1
B1	AS4-AS3-AS2
...	...
B4	AS4-AS3-AS2
C1	AS4-AS3
...	...
C3	AS4-AS3
D1	AS4
...	...
D4	AS4

D1 Table

Loop Prevention

The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a reach ability information, it checks to see if its autonomous system is in the path list to any destination. If it is, looping is involved and that network-path pair is discarded.

Policy Routing

Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.

Optimum Path

The optimum path is the path that fits the organization. Criteria may be: hop count, security and safety,reliability. Path vector routing can achieve optimum path by looking for a path best for the organization. Since all the AS are listed in the path.

BGP

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing. It first appeared in 1989 and has gone through four versions.

Types of Autonomous Systems

As we said before, the Internet is divided into hierarchical domains called autonomous systems (ASs). For example, a large corporation that manages its own network and has full control over it is an autonomous system. A local ISP that provides services to local customers is an autonomous system. Note that a single organization may choose to have multiple ASs because of geographical spread, different providers (ISPs), or even some local obstacles. We can divide autonomous systems into three categories: stub, multihomed, and transit.

Stub AS

A stub AS has only one connection to another AS. The interdomain data traffic in a stub AS can be either created or terminated in the AS. The hosts in the AS can send data traffic to other ASs. The hosts in the AS can receive data coming from hosts in other ASs. Data traffic, however, cannot pass through a stub AS. A stub AS is either a source or a sink. A good example of a stub AS is a small corporation or a small local ISP.

Multihomed AS

A multihomed AS has more than one connection to other ASs, but it is still only a source or sink for data traffic. It can receive data traffic from more than one AS. It can send data traffic to more than one AS, but there is no transient traffic. It does not allow data coming from one AS and going to another AS to pass through. A good example of a multihomed AS is a large corporation that is connected to more than one regional or national AS that does not allow transient traffic.

Transit AS

A transit AS is a multihomed AS that also allows transient traffic. Good examples of transit ASs are national and international ISPs (Internet backbones).

CIDR

BGP uses classless interdomain routing addresses. In other words, BGP uses a prefix, as discussed in Chapter 5, to define a destination address. The address and the number of bits (prefix length) are used in updating messages.

Path Attributes

In previous example, the path was presented as a list of AS . Actually, the path was presented as a list of attributes. The list of attributes help the receiving router make a better decision when applying its policy. Attributes are divided into two categories: well-known and optional.

Well-known: one that every BGP router should recognize

Mandatory: must appear in the description of a route

e.g., ORIGIN: the source of the routing information (RIP or OSPF)

e.g., AS_PATH: the list of AS through which the destination can be reached

e.g., NEXT_HOP: the next router to which data packet should be sent

Discretionary

Must be recognized by each router. But is not required to be included in every update message.

Optional: one that need not be recognized by every router

Transitive- Must be passed to the next router by the router that has not implemented this attribute

Nontransitive- One that should be discarded if the receiving router has not implemented it.

BGP Sessions

The exchange of routing information between two routers using BGP takes place in a session. A session is a connection that is established between two BGP routers only for the sake of exchanging routing information. To create a reliable environment, BGP uses the services of TCP. In other words, a session at the BGP level, as an application program, is a connection at the TCP level. However, there is a subtle difference between a connection in TCP made for BGP and other application programs. When a TCP connection is created for BGP, it can last for

a long time, until something unusual happens. For this reason, BGP sessions are sometimes referred to as semipermanent connections.

External and Internal BGP

If we want to be precise, BGP can have two types of sessions: external BGP (E-BGP) and internal BGP (I-BGP) sessions. The E-BGP session is used to exchange information between two speaker nodes belonging to two different autonomous systems. The IBGP session, on the other hand, is used to exchange routing information between two routers inside an autonomous system.

Types of Packets

BGP uses four different types of messages: open, update, keepalive, and notification.

Packet Format

All BGP packets share the same common header. Before studying the different types of packets, let us talk about this common header . The fields of this header are as follows:

Marker. The 16-byte marker field is reserved for authentication.

Length. This 2-byte field defines the length of the total message including the header.

Type. This 1-byte field defines the type of the packet. As we said before, we have four types, and the values 1 to 4 define those types.

Open Message

To create a neighborhood relationship, a router running BGP opens a TCP connection with a neighbor and sends an open message. If the neighbor accepts the neighborhood relationship, it responds with a keepalive message, which means that a relationship has been established between the two routers.

The fields of the open message are as follows:

Version. This 1-byte field defines the version of BGP. The current version is 4.

My autonomous system. This 2-byte field defines the autonomous system number.

Hold time. This 2-byte field defines the maximum number of seconds that can elapse until one of the parties receives a keepalive or update message from the other. If a router does not receive one of these messages during the hold time period, it considers the other party dead.

BGP identifier. This 4-byte field defines the router that sends the open message. The router usually uses one of its IP addresses (because it is unique) for this purpose.

Option length. The open message may contain some option parameters. In this case, this 1-byte field defines the length of the total option parameters. If there are no option parameters, the value of this field is zero.

Option parameters. If the value of the option parameter length is not zero, it means that there are some option parameters. Each option parameter itself has two subfields: the length of the parameter and the parameter value. The only option parameter defined so far is authentication.

Update Message

The update message is the heart of the BGP protocol. It is used by a router to withdraw destinations that have been advertised previously, announce a route to a new destination, or both. Note that BGP can withdraw several destinations that were advertised before, but it can only advertise one new destination in a single update message.

The update message fields are listed below:

Unfeasible routes length. This 2-byte field defines the length of the next field.

Withdrawn routes. This field lists all the routes that must be deleted from the previously advertised list.

Path attributes length. This 2-byte field defines the length of the next field.

Path attributes. This field defines the attributes of the path (route) to the network whose reachability is being announced in this message.

Network layer reachability information (NLRI). This field defines the network that is actually advertised by this message. It has a length field and an IP address prefix. The length defines the number of bits in the prefix. The prefix defines the common part of the network

address. For example, if the network is 153.18.7.0/24, the length of the prefix is 24 and the prefix is 153.18.7. BGP4 supports classless addressing and CIDR.

Keepalive Message

The routers (called peers in BGP parlance) running the BGP protocols exchange keepalive messages regularly (before their hold time expires) to tell each other that they are alive.

Notification Message

A notification message is sent by a router whenever an error condition is detected or a router wants to close the connection. The fields making up the notification message follow:

Error code. This 1-byte field defines the category of the error. See Table.

Error Code	*Error Code Description*	*Error Subcode Description*
1	Message header error	Three different subcodes are defined for this type of error: synchronization problem (1), bad message length (2), and bad message type (3).
2	Open message error	Six different subcodes are defined for this type of error: unsupported version number (1), bad peer AS (2), bad BGP identifier (3), unsupported optional parameter (4), authentication failure (5), and unacceptable hold time (6).
3	Update message error	Eleven different subcodes are defined for this type of error: malformed attribute list (1), unrecognized well-known attribute (2), missing well-known attribute (3), attribute flag error (4), attribute length error (5), invalid origin attribute (6), AS routing loop (7), invalid next hop attribute (8), optional attribute error (9), invalid network field (10), malformed AS_PATH (11).
4	Hold timer expired	No subcode defined.
5	Finite state machine error	This defines the procedural error. No subcode defined.
6	Cease	No subcode defined.

Error subcode. This 1-byte field further defines the type of error in each category.

Error data. This field can be used to give more diagnostic information about the error.

Encapsulation

BGP messages are encapsulated in TCP segments using the well-known port 179. This means that there is no need for error control and flow control. When a TCP connection is opened, the exchange of update, keepalive, and notification messages is continued until a notification message of type cease is sent.

UNIT-III

Multicast Routing – Multicast Routing Protocols. Group Management – IGMP Message – IGMP Operation – Process to Process Communication – UDP Operation – TCP Services - Flow Control.

Multicast Routing

In multicasting, there is one source and a group of destinations. The relationship is one to many. In this type of communication, the source address is a unicast address, but the destination address is a group address, a group of one or more destination networks in which there is at least one member of the group that is interested in receiving the multicast datagram. The group address defines the members of the group.

Multicast Routing Protocol is used to share information between routers to facilitate the transportation of IP multicast packets among networks. It formed the basis of the Internet's historic multicast backbone. A multicast router connected to a network is responsible to collect this type of information locally; the information collected can be globally propagated to other routers. The first task is done by the IGMP protocol; the second task is done by the multicast routing protocols. The Internet Group Management Protocol (IGMP) is responsible for correcting and interpreting information about group members in a network. It is one of the protocols designed at the IP layer for this purpose

Group Management

IGMP is not a multicasting routing protocol; it is a protocol that manages group membership. In any network, there are one or more multicast routers that distribute multicast packets to hosts or other routers. The IGMP protocol gives the multicast routers information about the membership status of hosts (routers) connected to the network. A multicast router may receive thousands of multicast packets every day for different groups. If a router has no knowledge about the

membership status of the hosts, it must forward all of these packets. This creates a lot of traffic and consumes bandwidth. A better solution is to keep a list of groups in the network for which there is at least one loyal member. IGMP helps the multicast router create and update this list

IGMP has gone through three versions. Versions 1 and 2 provide what is called any source multicast (ASM), which means that the group members receive a multicast message no matter where it comes from. The IGMP version 3 provides what is called sources pecific multicast (SSM), which means that the recipient can choose to receive multicast messages coming from a list of predefined sources. In this section we discuss only IGMPv3.

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

MEMBERSHIP QUERY

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.

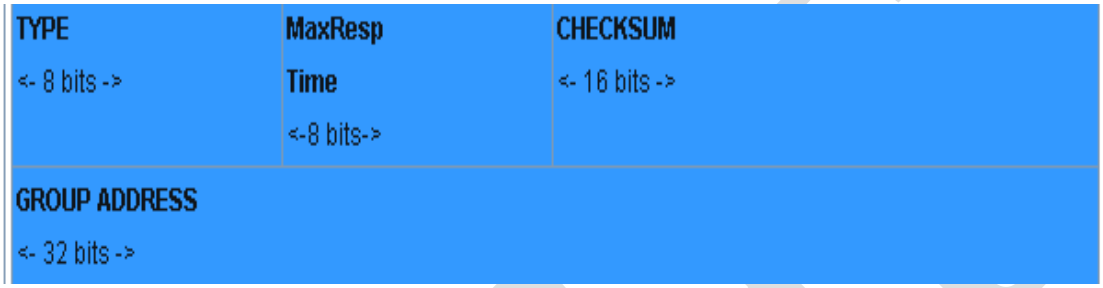
MEMBERSHIP REPORT

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

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.

IGMP Message Format



Type of IGMP message. There are three types: Membership Query, Membership Report and Leave Group.

Maximum Response Time

This field is used only in Membership Query messages. This field is the maximum time a host is allowed to produce and send a Membership Report message after receiving a Membership Query message.

Checksum

This is the one's compliment of the one's complement sum of the entire IGMP message, which basically works out to be the entire payload of the IP datagram the IGMP datagram is encapsulated within.

Group Address

Behavior of this field varies by the type of message sent:

Membership Query: (set to)

General Query: All zeroes

Group Specific Query: multicast group address

Membership Report: multicast group address

Leave Group: multicast group address

IGMP Operation

1. An IGMP-enabled router sends out several General Membership Queries at startup.
2. Hosts that are members of specific multicast groups send Membership Reports back to the router to report their membership.
3. The router receives the Membership Reports and builds lists of multicast group memberships for each attached network.
4. The router sets an interval for each group's updates and sends a Group-Specific Membership Query with this information in the Maximum Response Time field.
5. If router 'A' hears a query from another router (router 'B') on the same attached network, and router 'B' has a lower IP address, router 'A' will become a Non-Querier for that network. If, after a certain interval, router 'A' does not hear from router 'B' (the current Querier), router 'A' assumes the Querier is down and becomes the Querier for that network.

HOST JOINS MULTICAST GROUP

When a host joins a multicast group, it sends a 'Membership Report' to the router. The router makes an entry in it's table and forwards the membership report message.

HOST LEAVES MULTICAST GROUP

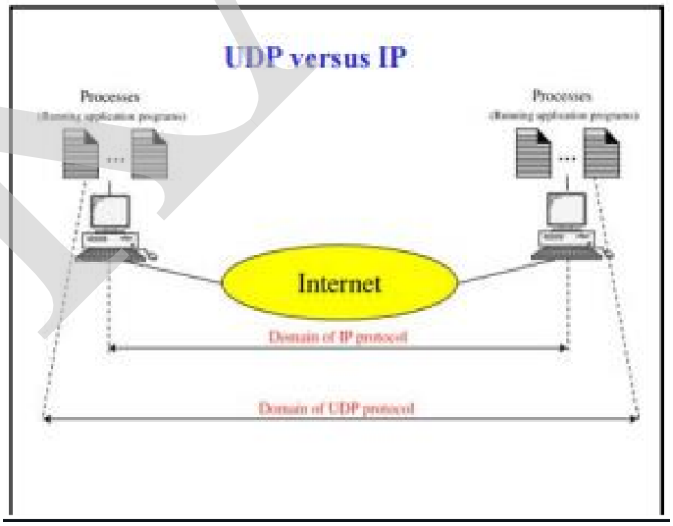
When a host leaves a multicast group, it sends a Leave Message to the router. The Querier router may, or may not send out a Group Specific Membership Query, based on whether or not the leaving host was the last host in the group.

Process to Process Communication

UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss tolerating connections between applications on the Internet. Both UDP and TCP run on top of the Internet Protocol (IP) and are sometimes referred to as UDP/IP or TCP/IP. UDP provides process-to-

process communication using sockets, a combination of IP addresses and port numbers. Several port numbers used by UDP are shown in Table .

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	Domain	Domain Name Service (DNS)
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)



ICANN Ranges

The ICANN has divided the port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports.

The Well Known Ports are those from 0 through 1023. DCCP Well Known ports SHOULD NOT be used without IANA registration.

Registered Ports are those from 1024 through 49151 DCCP Registered ports SHOULD NOT be used without IANA registration.

The Dynamic and/or Private Ports are those from 49152 through 65535

UDP Operation

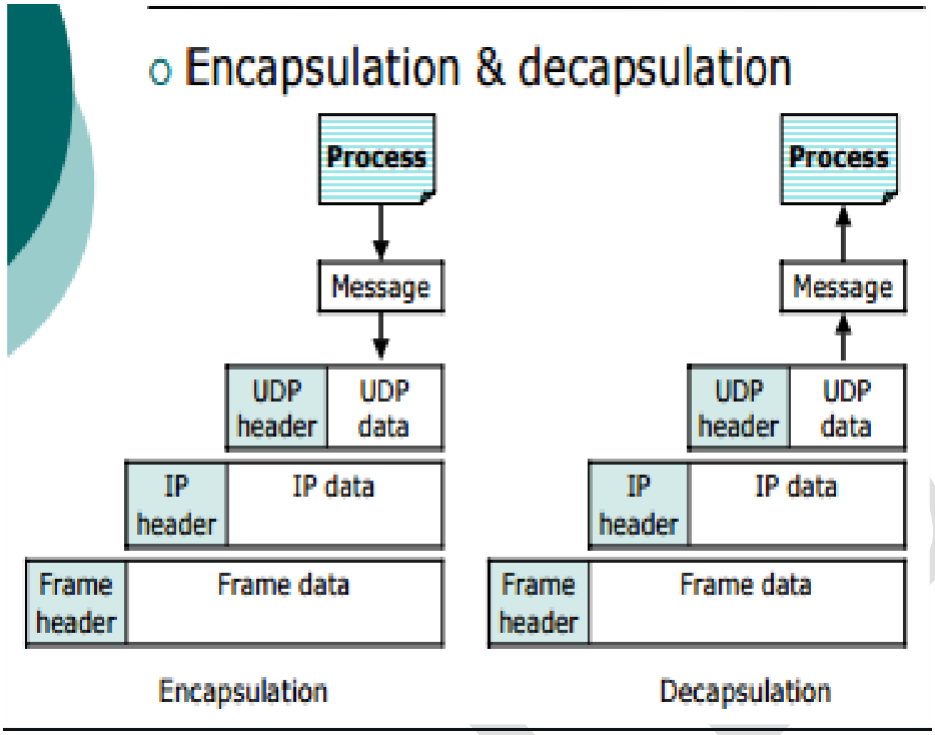
Connectionless services

Each datagram sent by UDP is an independent datagram. UDP cannot chop a stream of data into different related user datagram's . Each request must be small enough to fit into one user datagram. Only processes sending short message should use UDP.

Flow and error control

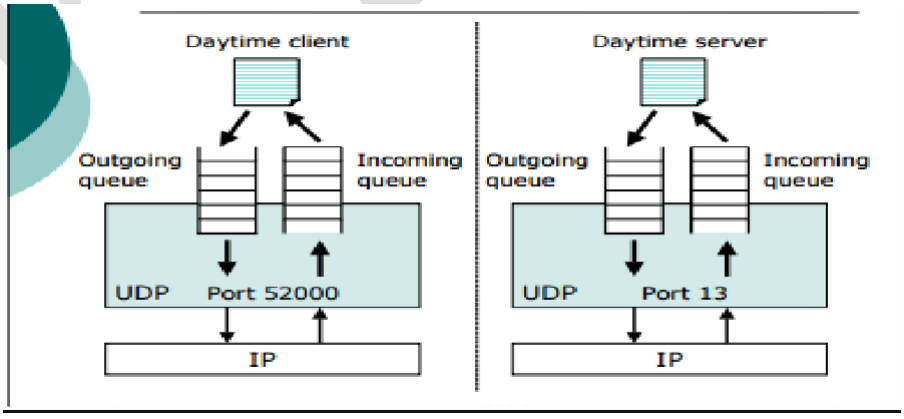
No flow control z No error control except for checksum . The sender does not know whether the message has been lost . When the receiver detects an error through checksum, it discards it silently . The process using UDP should provide flow and error control by itself.

Encapsulation & Decapsulation



Queuing

Queues are opened for server / client processes. 2 queues for each process. Incoming queue: receive messages . Outgoing queue: send messages. The queues function as long as the process is running . The queues are destroyed when the process terminates.



Queues on the client side

The client process requests a port number from the operating system . The process opens incoming and outgoing queues with the requested port number

Queues on the server side

The server asks for incoming and outgoing queues using its well-known port number.

Outgoing queue overflow

The operating system asks the server / client to wait before sending any more messages.

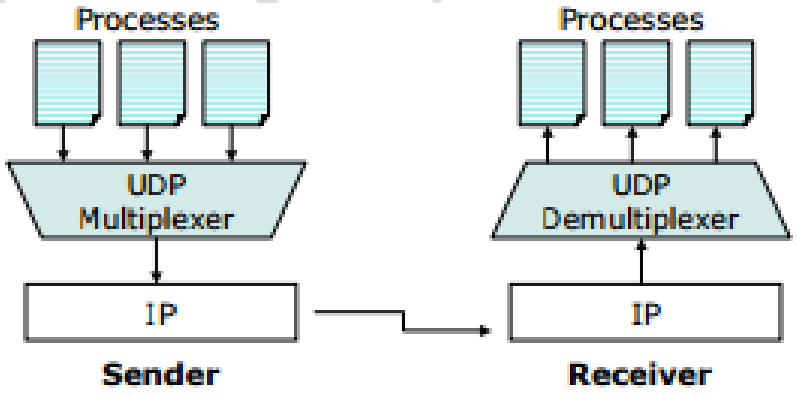
Incoming queue overflow

UDP drops the datagram and asks the ICMP protocol to send port unreachable message to the datagram sender.

No incoming queue created for the port number specified in the arrived datagramUDP discards the datagram and asks the ICMP protocol to send port unreachable message to the datagram sender.

Multiplexing & demultiplexing

In a host running TCP/IP, there are: One UDP and several processes that want to use UDP services.



Multiplexing

Sender side: there may be several processes that need to send user datagrams

Many-to-one relationship: multiplexing - UDP accepts messages from different processes . Differentiates messages by their port numbers . Adds header to each message. Passes user datagram to IP

Demultiplexing

Receiver side: there may be several processes that can receive user datagrams

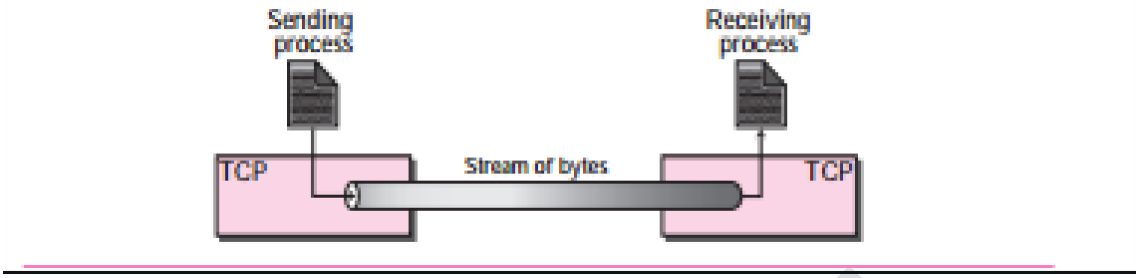
One-to-many relationship: demultiplexing - UDP receives user datagram from IP . Checks errors in user datagram . Drops the header . Delivers the message to the appropriate process based on the port number.

TCP Services

TCP lies between the application layer and the network layer, and serves as the intermediary between the application programs and the network operations. Before discussing TCP in detail, let us see the services offered by TCP to the processes at the application layer. Process-to-Process Communication .As with UDP, TCP provides process-to-process communication using port numbers.

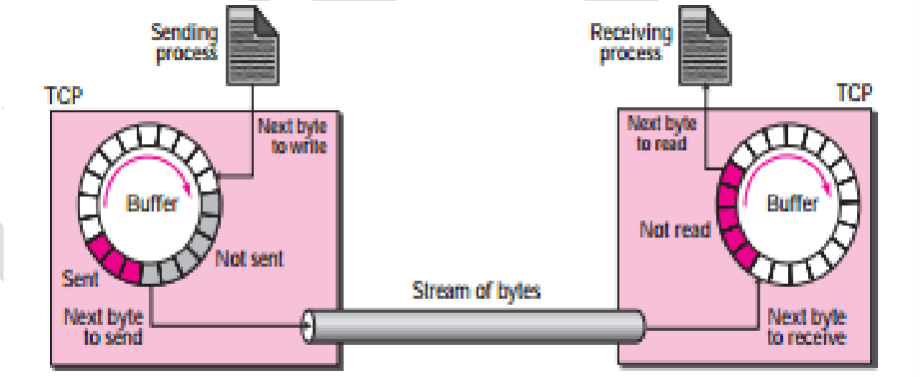
Stream Delivery

Service TCP, unlike UDP, is a stream-oriented protocol. In UDP, a process sends messages with predefined boundaries to UDP for delivery. UDP adds its own header to each of these messages 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 processes seem to be connected by an imaginary “tube” that carries their bytes across the Internet. This imaginary environment is depicted in Figure . The sending process produces (writes to) the stream of bytes and the receiving process consumes (reads from) them.



Sending and Receiving Buffers

Because the sending and the receiving processes may not necessarily write or read data at the same rate, 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 mechanisms used by TCP. One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure. For simplicity, we have shown two buffers of 20 bytes each; normally the buffers 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.

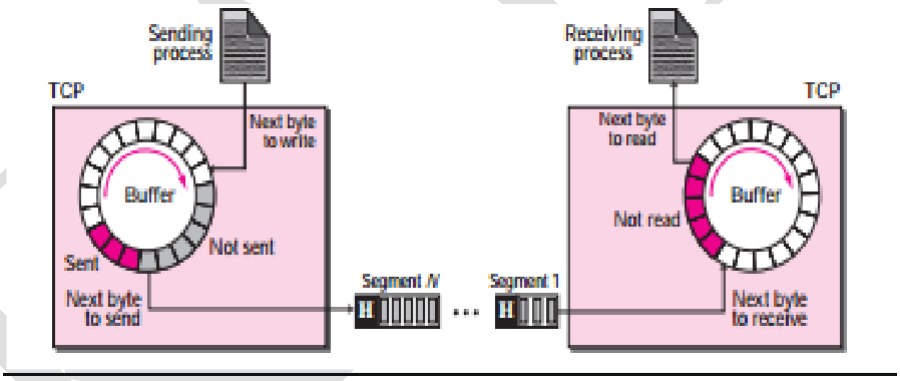


At the sender, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer). The colored area holds bytes that have been sent but not yet acknowledged. The TCP sender keeps these bytes in the buffer until it receives an acknowledgment. The shaded area contains bytes to be sent by the sending TCP. two areas (shown as white and colored). The white area contains empty chambers to be filled by bytes received from the network. The colored sections contain received bytes that can be read

by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

Segments

Although buffering handles the disparity between the speed of the producing and consuming processes, 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 a segment. TCP adds a header to each segment (for control purposes) and delivers the segment to the IP layer for 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 the TCP sender with the receiving application process unaware of TCP's activities. Figure shows how segments are created from the bytes in the buffers.



Full-Duplex Communication

TCP offers full-duplex service, where data can flow in both directions at the same time. Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

Multiplexing and Demultiplexing

Like UDP, TCP performs multiplexing at the sender and demultiplexing at the receiver. However, since TCP is a connection-oriented protocol, a connection needs to be established for each pair of processes.

Connection-Oriented Service

TCP, unlike UDP, is a connection-oriented protocol. As shown in Chapter 13, when a process at site A wants to send to and receive data from another process at site B, the following three phases occur:

1. The two TCPs establish a virtual 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.

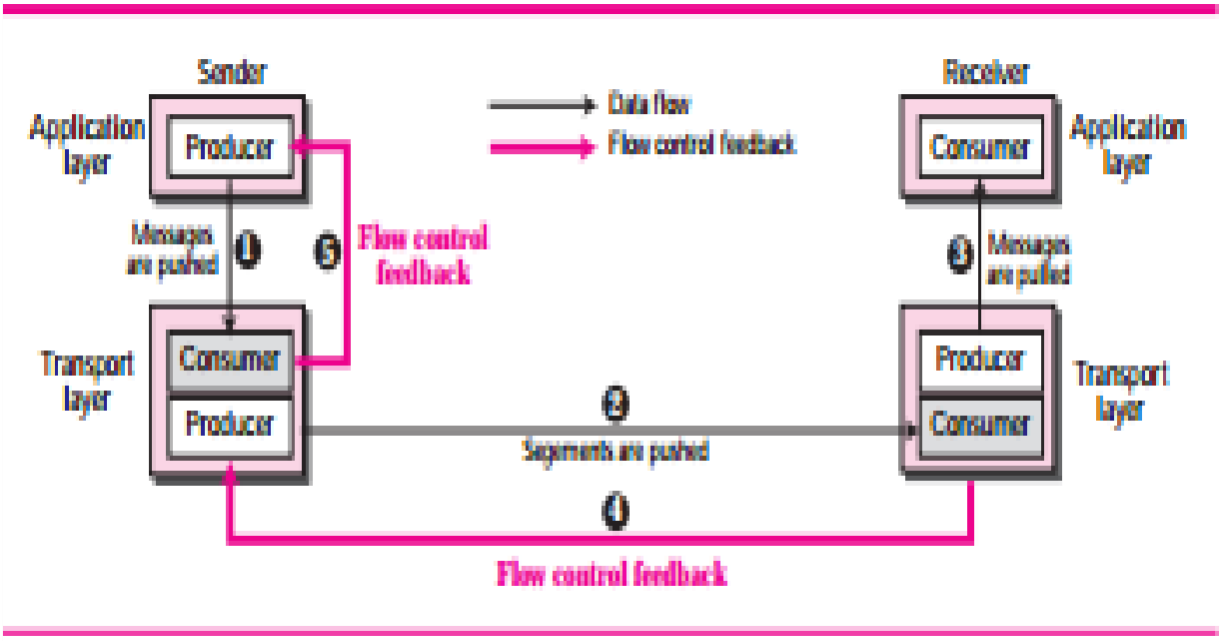
The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost, or corrupted, and then resent. Each may be routed over a different path to reach the destination. There is no physical connection. TCP creates a stream-oriented environment in which it accepts the responsibility of delivering the bytes in order to the other site.

Reliable Service

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

Flow Control.

Flow control balances the rate a producer creates data with the rate a consumer can use the data. TCP separates flow control from error control. Figure shows unidirectional data transfer between a sender and a receiver; bidirectional data transfer can be deduced from unidirectional one.



The figure shows that data travel from the sending process down to the sending TCP, from the sending TCP to the receiving TCP, and from receiving TCP up to the receiving process (paths 1, 2, and 3). Flow control feedbacks, however, are traveling from the receiving TCP to the sending TCP and from the sending TCP up to the sending process (paths 4 and 5). Most implementations of TCP do not provide flow control feedback from the receiving process to the receiving TCP; they let the receiving process pull data from the receiving TCP whenever it is ready to do so. In other words, the receiving TCP controls the sending TCP; the sending TCP controls the sending process.

Opening and Closing Windows

To achieve flow control, TCP forces the sender and the receiver to adjust their window sizes, although the size of the buffer for both parties is fixed when the connection is established. The receive window closes (moves its left wall to the right) when more bytes arrive from the sender; it opens (moves its right wall to the right) when more bytes are pulled by the process. We assume that it does not shrink (the right wall does not move to the left). The opening, closing,

and shrinking of the send window is controlled by the receiver. The send window closes (moves its left wall to the right) when a new acknowledgement allows it to do so. The send window opens (its right wall moves to the right) when the receive window size (rwnd) advertised by the receiver allows it to do so. The send window shrinks on occasion. We assume that this situation does not occur.

Sliding window

The TCP sliding window determines the number of unacknowledged bytes, *x*, that one system can send to another. Two factors determine the value of *x*:

- The size of the send buffer on the sending system
- The size and available space in the receive buffer on the receiving system

The sending system cannot send more bytes than space that is available in the receive buffer on the receiving system. TCP on the sending system must wait to send more data until all bytes in the current send buffer are acknowledged by TCP on the receiving system.

On the receiving system, TCP stores received data in a receive buffer. TCP acknowledges receipt of the data, and *advertises* (communicates) a new *receive window* to the sending system. The receive window represents the number of bytes that are available in the receive buffer. If the receive buffer is full, the receiving system advertises a receive window size of zero, and the sending system must wait to send more data. After the receiving application retrieves data from the receive buffer, the receiving system can then advertise a receive window size that is equal to the amount of data that was read. Then, TCP on the sending system can resume sending data.
The available space in the receive buffer depends on how quickly data is read from the buffer by the receiving application. TCP keeps the data in its receive buffer until the receiving application reads it from that buffer. After the receiving application reads the data, that space in the buffer is available for new data.

POSSIBLE QUESTIONS

Two Marks

1. What is sliding window?
2. Differentiate unicast and multicast.
3. Define distance vector routing.
4. Mention the purpose of RIP
5. What is the use of LSP in link state protocol?
6. Define BGP.
7. What is group management?
8. Mention the fields in UDP.
9. Define flow control.
10. Mention the types of IGMP messages.

Eight Marks

1. Write about the operation of distance vector routing.
2. Give a brief description about RIP message format.
3. Discuss about building of routing table and formation of shortest path tree in link state routing.
4. Discuss about links in OSPF.
5. Explain path vector routing with an example.
6. Write about BGP sessions and packet types.
7. Discuss about multi cast routing and its applications.
8. Write a brief note on IGMP operation.
9. Explain with an example process to process communication.
10. Discuss about UDP operation.
11. Dicsuss about flow control with an example.

UNIT-III

Multicast Routing – Multicast Routing Protocols. Group Management – IGMP Message – IGMP Operation – Process to Process Communication – UDP Operation – TCP Services - Flow Control.

Multicast Routing

In multicasting, there is one source and a group of destinations. The relationship is one to many. In this type of communication, the source address is a unicast address, but the destination address is a group address, a group of one or more destination networks in which there is at least one member of the group that is interested in receiving the multicast datagram. The group address defines the members of the group.

Multicast Routing Protocol is used to share information between routers to facilitate the transportation of IP multicast packets among networks. It formed the basis of the Internet's historic multicast backbone. A multicast router connected to a network is responsible to collect this type of information locally; the information collected can be globally propagated to other routers. The first task is done by the IGMP protocol; the second task is done by the multicast routing protocols. The Internet Group Management Protocol (IGMP) is responsible for correcting and interpreting information about group members in a network. It is one of the protocols designed at the IP layer for this purpose

Group Management

IGMP is not a multicasting routing protocol; it is a protocol that manages group membership. In any network, there are one or more multicast routers that distribute multicast packets to hosts or other routers. The IGMP protocol gives the multicast routers information about the membership status of hosts (routers) connected to the network. A multicast router may receive thousands of multicast packets every day for different groups. If a router has no knowledge about the

membership status of the hosts, it must forward all of these packets. This creates a lot of traffic and consumes bandwidth. A better solution is to keep a list of groups in the network for which there is at least one loyal member. IGMP helps the multicast router create and update this list

IGMP has gone through three versions. Versions 1 and 2 provide what is called any source multicast (ASM), which means that the group members receive a multicast message no matter where it comes from. The IGMP version 3 provides what is called sources pecific multicast (SSM), which means that the recipient can choose to receive multicast messages coming from a list of predefined sources. In this section we discuss only IGMPv3.

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

MEMBERSHIP QUERY

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.

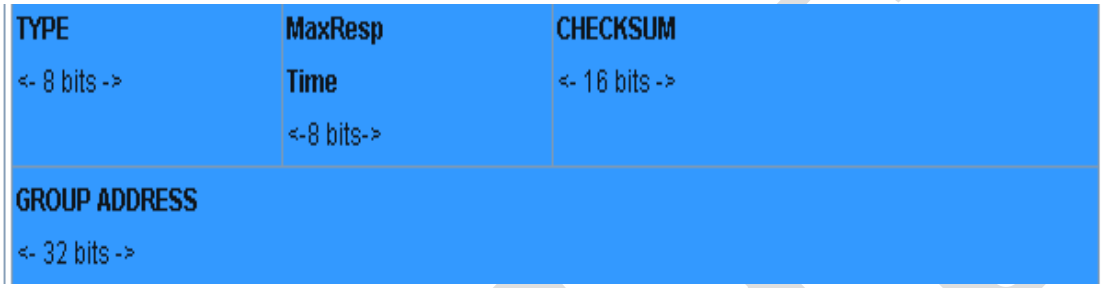
MEMBERSHIP REPORT

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

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.

IGMP Message Format



Type of IGMP message. There are three types: Membership Query, Membership Report and Leave Group.

Maximum Response Time

This field is used only in Membership Query messages. This field is the maximum time a host is allowed to produce and send a Membership Report message after receiving a Membership Query message.

Checksum

This is the one's compliment of the one's complement sum of the entire IGMP message, which basically works out to be the entire payload of the IP datagram the IGMP datagram is encapsulated within.

Group Address

Behavior of this field varies by the type of message sent:

Membership Query: (set to)

General Query: All zeroes

Group Specific Query: multicast group address

Membership Report: multicast group address

Leave Group: multicast group address

IGMP Operation

1. An IGMP-enabled router sends out several General Membership Queries at startup.
2. Hosts that are members of specific multicast groups send Membership Reports back to the router to report their membership.
3. The router receives the Membership Reports and builds lists of multicast group memberships for each attached network.
4. The router sets an interval for each group's updates and sends a Group-Specific Membership Query with this information in the Maximum Response Time field.
5. If router 'A' hears a query from another router (router 'B') on the same attached network, and router 'B' has a lower IP address, router 'A' will become a Non-Querier for that network. If, after a certain interval, router 'A' does not hear from router 'B' (the current Querier), router 'A' assumes the Querier is down and becomes the Querier for that network.

HOST JOINS MULTICAST GROUP

When a host joins a multicast group, it sends a 'Membership Report' to the router. The router makes an entry in it's table and forwards the membership report message.

HOST LEAVES MULTICAST GROUP

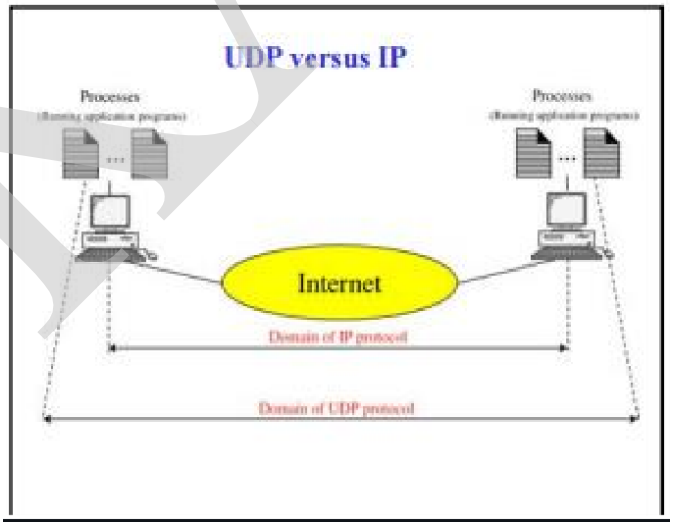
When a host leaves a multicast group, it sends a Leave Message to the router. The Querier router may, or may not send out a Group Specific Membership Query, based on whether or not the leaving host was the last host in the group.

Process to Process Communication

UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss tolerating connections between applications on the Internet. Both UDP and TCP run on top of the Internet Protocol (IP) and are sometimes referred to as UDP/IP or TCP/IP. UDP provides process-to-

process communication using sockets, a combination of IP addresses and port numbers. Several port numbers used by UDP are shown in Table .

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	Domain	Domain Name Service (DNS)
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)



ICANN Ranges

The ICANN has divided the port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports.

The Well Known Ports are those from 0 through 1023. DCCP Well Known ports SHOULD NOT be used without IANA registration.

Registered Ports are those from 1024 through 49151 DCCP Registered ports SHOULD NOT be used without IANA registration.

The Dynamic and/or Private Ports are those from 49152 through 65535

UDP Operation

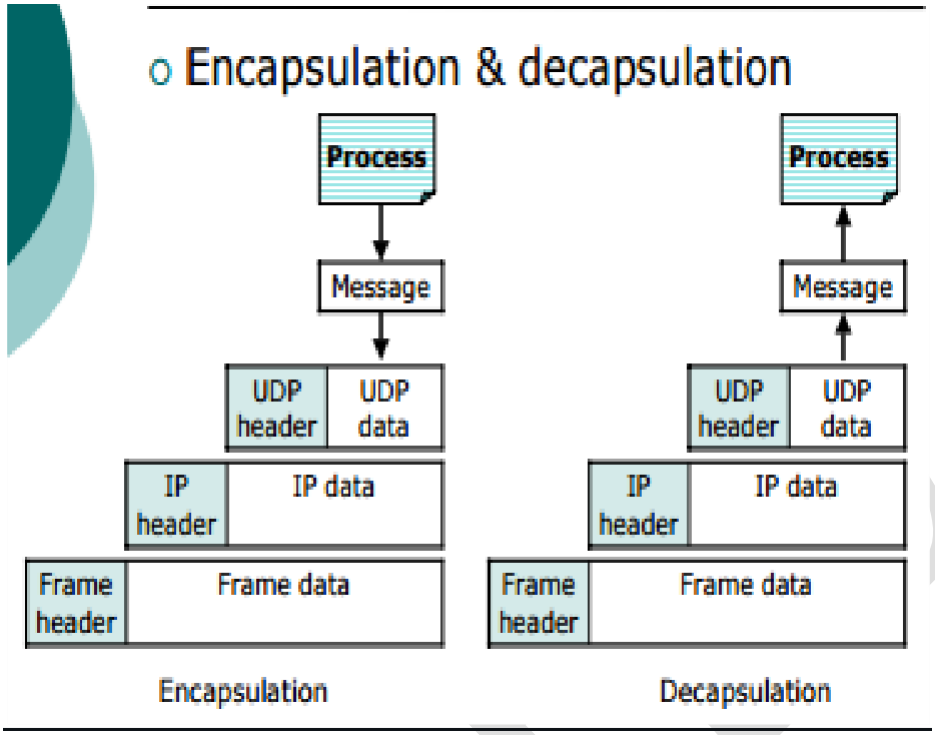
Connectionless services

Each datagram sent by UDP is an independent datagram. UDP cannot chop a stream of data into different related user datagram's . Each request must be small enough to fit into one user datagram. Only processes sending short message should use UDP.

Flow and error control

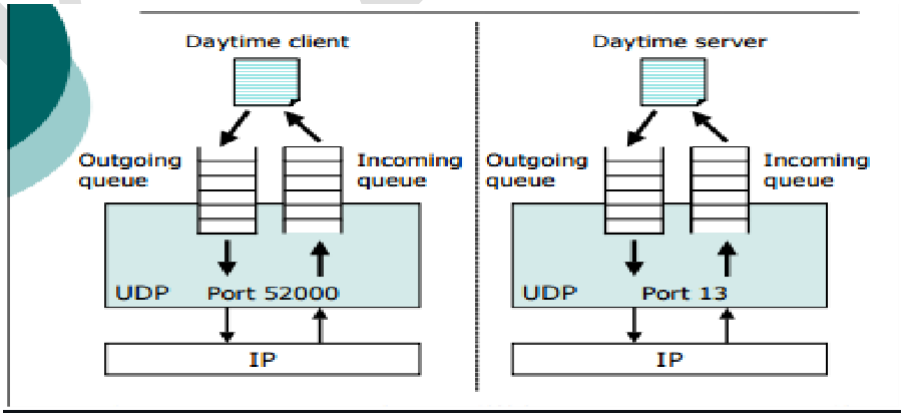
No flow control z No error control except for checksum . The sender does not know whether the message has been lost . When the receiver detects an error through checksum, it discards it silently . The process using UDP should provide flow and error control by itself.

Encapsulation & Decapsulation



Queuing

Queues are opened for server / client processes. 2 queues for each process. Incoming queue: receive messages . Outgoing queue: send messages. The queues function as long as the process is running . The queues are destroyed when the process terminates.



Queues on the client side

The client process requests a port number from the operating system . The process opens incoming and outgoing queues with the requested port number

Queues on the server side

The server asks for incoming and outgoing queues using its well-known port number.

Outgoing queue overflow

The operating system asks the server / client to wait before sending any more messages.

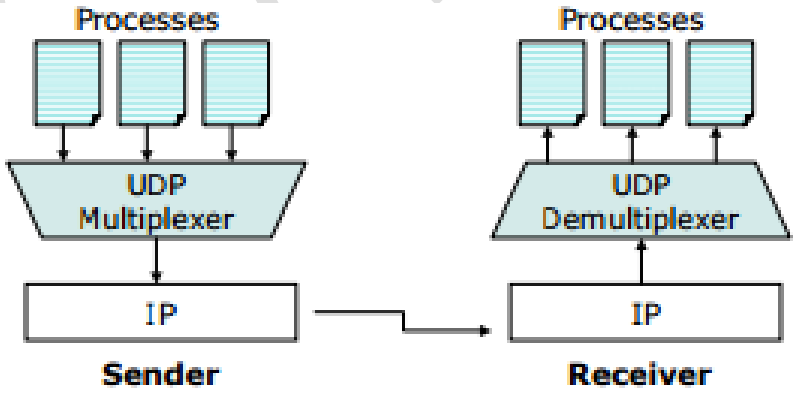
Incoming queue overflow

UDP drops the datagram and asks the ICMP protocol to send port unreachable message to the datagram sender.

No incoming queue created for the port number specified in the arrived datagramUDP discards the datagram and asks the ICMP protocol to send port unreachable message to the datagram sender.

Multiplexing & demultiplexing

In a host running TCP/IP, there are: One UDP and several processes that want to use UDP services.



Multiplexing

Sender side: there may be several processes that need to send user datagrams

Many-to-one relationship: multiplexing - UDP accepts messages from different processes . Differentiates messages by their port numbers . Adds header to each message. Passes user datagram to IP

Demultiplexing

Receiver side: there may be several processes that can receive user datagrams

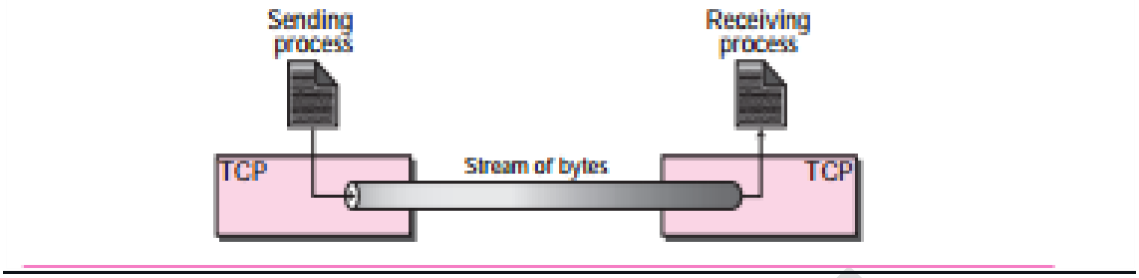
One-to-many relationship: demultiplexing - UDP receives user datagram from IP . Checks errors in user datagram . Drops the header . Delivers the message to the appropriate process based on the port number.

TCP Services

TCP lies between the application layer and the network layer, and serves as the intermediary between the application programs and the network operations. Before discussing TCP in detail, let us see the services offered by TCP to the processes at the application layer. Process-to-Process Communication .As with UDP, TCP provides process-to-process communication using port numbers.

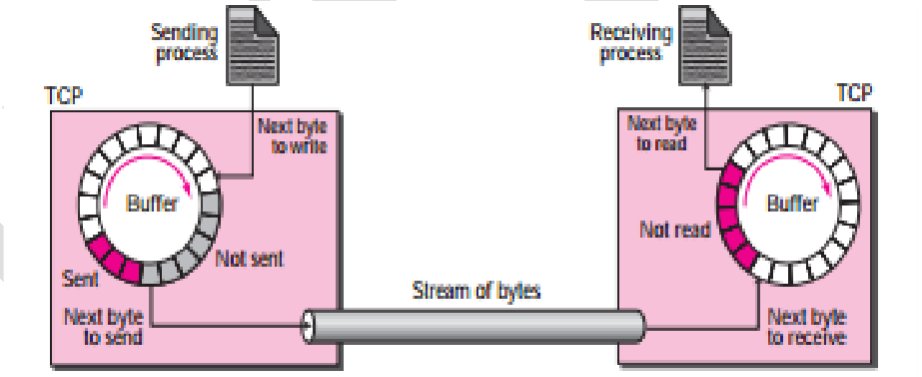
Stream Delivery

Service TCP, unlike UDP, is a stream-oriented protocol. In UDP, a process sends messages with predefined boundaries to UDP for delivery. UDP adds its own header to each of these messages 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 processes seem to be connected by an imaginary “tube” that carries their bytes across the Internet. This imaginary environment is depicted in Figure . The sending process produces (writes to) the stream of bytes and the receiving process consumes (reads from) them.



Sending and Receiving Buffers

Because the sending and the receiving processes may not necessarily write or read data at the same rate, 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 mechanisms used by TCP. One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure. For simplicity, we have shown two buffers of 20 bytes each; normally the buffers 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.

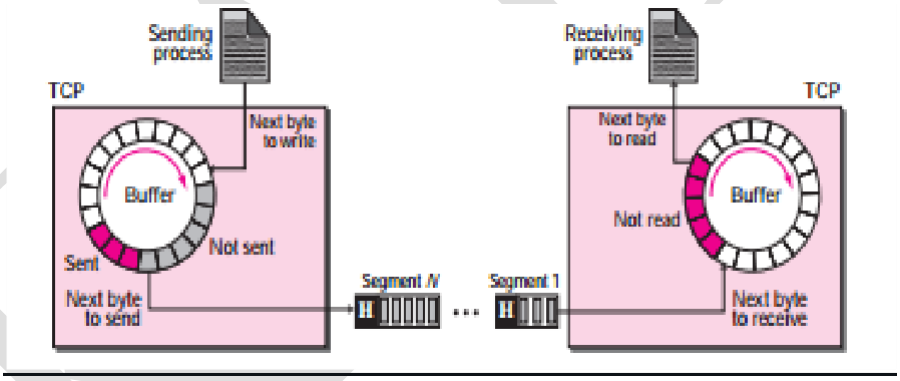


At the sender, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer). The colored area holds bytes that have been sent but not yet acknowledged. The TCP sender keeps these bytes in the buffer until it receives an acknowledgment. The shaded area contains bytes to be sent by the sending TCP. two areas (shown as white and colored). The white area contains empty chambers to be filled by bytes received from the network. The colored sections contain received bytes that can be read

by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

Segments

Although buffering handles the disparity between the speed of the producing and consuming processes, 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 a segment. TCP adds a header to each segment (for control purposes) and delivers the segment to the IP layer for 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 the TCP sender with the receiving application process unaware of TCP's activities. Figure shows how segments are created from the bytes in the buffers.



Full-Duplex Communication

TCP offers full-duplex service, where data can flow in both directions at the same time. Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

Multiplexing and Demultiplexing

Like UDP, TCP performs multiplexing at the sender and demultiplexing at the receiver. However, since TCP is a connection-oriented protocol, a connection needs to be established for each pair of processes.

Connection-Oriented Service

TCP, unlike UDP, is a connection-oriented protocol. As shown in Chapter 13, when a process at site A wants to send to and receive data from another process at site B, the following three phases occur:

1. The two TCPs establish a virtual 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.

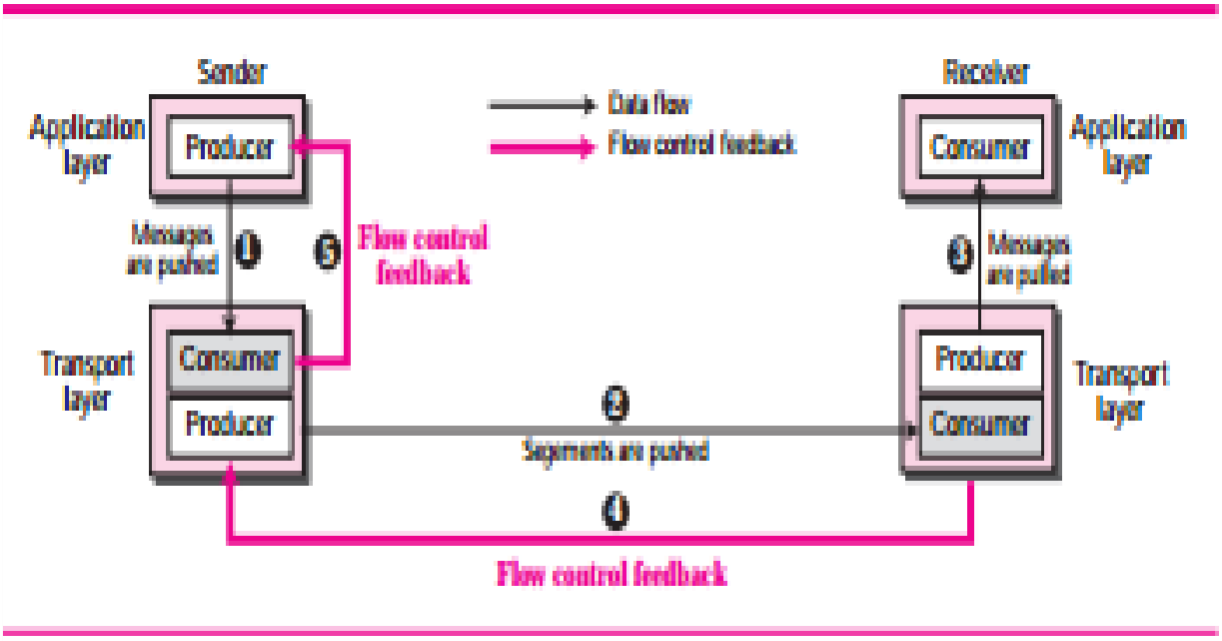
The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost, or corrupted, and then resent. Each may be routed over a different path to reach the destination. There is no physical connection. TCP creates a stream-oriented environment in which it accepts the responsibility of delivering the bytes in order to the other site.

Reliable Service

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

Flow Control.

Flow control balances the rate a producer creates data with the rate a consumer can use the data. TCP separates flow control from error control. Figure shows unidirectional data transfer between a sender and a receiver; bidirectional data transfer can be deduced from unidirectional one.



The figure shows that data travel from the sending process down to the sending TCP, from the sending TCP to the receiving TCP, and from receiving TCP up to the receiving process (paths 1, 2, and 3). Flow control feedbacks, however, are traveling from the receiving TCP to the sending TCP and from the sending TCP up to the sending process (paths 4 and 5). Most implementations of TCP do not provide flow control feedback from the receiving process to the receiving TCP; they let the receiving process pull data from the receiving TCP whenever it is ready to do so. In other words, the receiving TCP controls the sending TCP; the sending TCP controls the sending process.

Opening and Closing Windows

To achieve flow control, TCP forces the sender and the receiver to adjust their window sizes, although the size of the buffer for both parties is fixed when the connection is established. The receive window closes (moves its left wall to the right) when more bytes arrive from the sender; it opens (moves its right wall to the right) when more bytes are pulled by the process. We assume that it does not shrink (the right wall does not move to the left). The opening, closing,

and shrinking of the send window is controlled by the receiver. The send window closes (moves its left wall to the right) when a new acknowledgement allows it to do so. The send window opens (its right wall moves to the right) when the receive window size (rwnd) advertised by the receiver allows it to do so. The send window shrinks on occasion. We assume that this situation does not occur.

Sliding window

The TCP sliding window determines the number of unacknowledged bytes, *x*, that one system can send to another. Two factors determine the value of *x*:

- The size of the send buffer on the sending system
- The size and available space in the receive buffer on the receiving system

The sending system cannot send more bytes than space that is available in the receive buffer on the receiving system. TCP on the sending system must wait to send more data until all bytes in the current send buffer are acknowledged by TCP on the receiving system.

On the receiving system, TCP stores received data in a receive buffer. TCP acknowledges receipt of the data, and *advertises* (communicates) a new *receive window* to the sending system. The receive window represents the number of bytes that are available in the receive buffer. If the receive buffer is full, the receiving system advertises a receive window size of zero, and the sending system must wait to send more data. After the receiving application retrieves data from the receive buffer, the receiving system can then advertise a receive window size that is equal to the amount of data that was read. Then, TCP on the sending system can resume sending data.
The available space in the receive buffer depends on how quickly data is read from the buffer by the receiving application. TCP keeps the data in its receive buffer until the receiving application reads it from that buffer. After the receiving application reads the data, that space in the buffer is available for new data.

POSSIBLE QUESTIONS

Two Marks

1. What is sliding window?
2. Differentiate unicast and multicast.
3. Define distance vector routing.
4. Mention the purpose of RIP
5. What is the use of LSP in link state protocol?
6. Define BGP.
7. What is group management?
8. Mention the fields in UDP.
9. Define flow control.
10. Mention the types of IGMP messages.

Eight Marks

1. Write about the operation of distance vector routing.
2. Give a brief description about RIP message format.
3. Discuss about building of routing table and formation of shortest path tree in link state routing.
4. Discuss about links in OSPF.
5. Explain path vector routing with an example.
6. Write about BGP sessions and packet types.
7. Discuss about multi cast routing and its applications.
8. Write a brief note on IGMP operation.
9. Explain with an example process to process communication.
10. Discuss about UDP operation.
11. Dicsuss about flow control with an example.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

ONE MARK QUESTIONS

DEPARTMENT OF CS, CA & IT

STAFF NAME: S.MANJU PRIYA **SUBJECT NAME: INTERNETWORKING WITH TCP/IP**

SUB.CODE: 18CSP201 **UNIT III** **SEMESTER: II**

S.NO	Question	Choice1	Choice2	Choice3	Choice4	Ans
1	_____ is used to measure distance	Core	Vector	Dynamic	Hops	Hops
2	___is a group of network controlled by single administrative autthority	Non autonomous system	Autonomous system	Single system	hybrid system	Autonomous system
3	_____ forms the central interconnect for an internet	Base header	Application Gateway	Backbone network	Local Area Network	Backbone network
4	_____ is used to pass information between two autonomous system	EGP	IGP	DGP	CGP	EGP
5	BGP evolves _____ versions	3	2	4	1	4
6	__ protocol is used to propagate routing information inside a autonomous system	IGP	RIP	RFC	RAP	RIP

7	RIP messages can be broadly classified into _____ types	3	2	4	5	2
8	_____ reference to processor choosing the path to send packets	Router	Hub	Bridge	switch	Router
9	Routers exchange _____ to initialize their network	OSFP	OSPF	OPSF	OFSP	OSPF
10	OSPF sends _____ message to test neighbour reach ability	Hello	Check	Transmit	receipt	Hello
11	Command1 terns on the _____ mode	update	Request	Trace	receipt	Trace
12	Final technique for solving the slow convengaice problem is called _____	Triggered updates	Poison reverse	Both a and b	Null	Poison reverse
13	Path information in BGP allows the receiver to check for _____	Routing loops	Routing algorithms	Both A and B	Routing protocols	Routing loops
14	BGP periodically exchange ____message to text network connectivity	OPEN	EXCHANGE	CHECK	KEEPALIVE	KEEPALIVE
15	Command 10 in RIP is _____	Update Request	Update Response	Update acknowledgement	Update Receipt	Update Response
16	_____ used by routing protocols to combine multiple destination same hope	Route aggregation	Route alert	Route integration	Route dismiss	Route aggregation
17	The _____is one of the necessary protocols that is involved In multicasting	ICMP	IGMP	TCP	IP	IGMP
18	_____ 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
19	To prevent unnecessary traffic , IGMP uses a _____ strategy	Maximum response time	Minimum response time	Delayed response	response	Delayed response

20	Well known port numbers are less than_____	255	1024	125	206	1024
21	_____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
22	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
23	TCP offers _____ services	Full duplex	Half Duplex	Both a and b	Multi duplex	Full duplex
24	A packet in a TCP is called _____	Frame	Segment	Data	Bits	Segment
25	_____ 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
26	To accomplish flow control, TCP uses a _____ protocol	ICMP	IGMP	Sliding window	SNMP	Sliding window
27	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
28	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
29	_____ protocol is a group shared protocol that uses a core as the root of the tree	ICMP	DVMRP	CBT protocol	IGMP	CBT protocol
30	PIM-SM is used in a sparse multicast environment such as _____	WAN	LAN	MAN	Both a and b	WAN
31	The value of group address is _____ for a general query message	2	3	0	4	0
32	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

33	The UDP packets, called used data grams have a fixed size header of _____ bytes	6	8	32	64	8
34	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
35	SYN flooding attack belongs to a group of security attacks known as _____ attack	Denial of service	Mosquerade	Replay	Null	Denial of service
36	One of the algorithm used in TCP congestion control is _____	Fast start	Slow start	RFC algorithm	super fast	Slow start
37	_____ 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
38	In destination unreachable error reporting msg _____represent a protocol is unreachable.	code1	code 2	code 3	code 4	code 2
39	In service type TOS bit 0001 represent _____	Normal	Minimize Cost	Maximum reliability	Minimize delay	Minimize Cost
40	_____ is not a multicasting routing protocol.	ICMP	IGMP	TCP	TCP/IP	IGMP
41	IGMP stands for	Internet Group Management Protocol	Internet Group Maintenance	Information Group Management	Information Group Maintenance	Internet Group Management Protocol
42	_____ managers group membership	ICMP	IGMP	TCP	TCP/IP	IGMP
43	_____ is called a connectionless, unreliable transport protocol.	UDP	TCP	SCTP	FTP	UDP
44	UDP means _____	User Datagram Protocol	User Defined Protocol	User Derived Protocol	All the above	User Datagram Protocol
45	In Multicasting _____process multicast packet are encapsulated network.	Tunneling	Trimming	Transporting	Threading	Tunneling

46	UDP provides _____ service.	connection-oriented	connectionless	fast connection	slow connection	connectionless
47	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
48	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
49	When client process has no more data to send issues an	active close	passive close	full close	half close	active close
50	In ____ protocol host uses a window for outbound communication.	UDP	FTP	Sliding window protocol	SMTP	Sliding window protocol
51	RTO means _____	Remote Time out	Retransmission Time in	Retransmission Time in	Remote timing	Retransmission Time in
52	One of the algorithms used in TCP congestion control is _____	Fast Start	Fast Stop	Slow Start	Slow stop	Slow Start
53	_____ defines the size of the buffer in the local TCP	Hardware Type	Protocol size	Software Size	Buffer Size	Buffer Size
54	_____ Protocol returns the Quote of the Day	Quote	Daytime	Users	Discard.	Quote
55	_____ address is used for Multicasting.	Class B	Class D	Class A	Class E	Class D
56	_____ address is used for future purpose	Class B	Class D	Class A	Class E	Class E
57	_____ Address defines a group of computers.	Unicast	Multicast	Broadcast	nocast	Multicast

UNIT-IV

SYLLABUS

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

BOOTP

The Bootstrap Protocol (BOOTP) is the prerunner of DHCP. It is a client/server protocol designed to overcome the two deficiencies of the RARP protocol. First, since it is a client/server program, the BOOTP server can be anywhere in the Internet can provide all pieces of information we mentioned above, including the IP address. To provide the four pieces of information described above, it removes all restriction about the RARP protocol. BOOTP, however, is a static configuration protocol. When a client requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address. This implies that the binding between the physical address and the IP address of the client already exists. The binding is predetermined. There are some situations in which we need a dynamic configuration protocol. For example, when a host moves from one physical network to another, its physical address changes. As another example, there are occasions when a host wants a temporary IP address to be used for a period of time. 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. As we will see shortly, DHCP has been devised to handle these shortcomings.

DHCP

The Dynamic Host Configuration Protocol (DHCP) is a client/server protocol designed to provide the four pieces of information for a diskless computer or a computer that is booted for the first time. DHCP is a successor to BOOTP and is backward compatible with it. Although

BOOTP is considered deprecated, there may be some systems that may still use BOOTP for host configuration. The part of the discussion in this chapter that does not deal with the dynamic aspect of DHCP can also be applied to BOOTP.

DHCP operation

The DHCP client and server can either be on the same network or on different networks. Let us discuss each situation separately. Same Network Although the practice is not very common, the administrator may put the client and the server on the same network.

In this case, the operation can be described as follows:

1. The DHCP server issues a passive open command on UDP port number 67 and waits for a client.

2. A booted client issues an active open command on port number 68 (this number will be explained later). The message is encapsulated in a UDP user datagram, using the destination port number 67 and the source port number 68. The UDP user datagram, in turn, is encapsulated in an IP datagram. The reader may ask how a client can send an IP datagram when it knows neither its own IP address (the source address) nor the server's IP address (the destination address). The client uses all 0s as the source address and all 1s as the destination address.

3. The server responds with either a broadcast or a unicast message using UDP source port number 67 and destination port number 68. The response can be unicast because the server knows the IP address of the client. It also knows the physical address of the client, which means it does not need the services of ARP for logical to physical address mapping. However, some systems do not allow the bypassing of ARP, resulting in the use of the broadcast address.

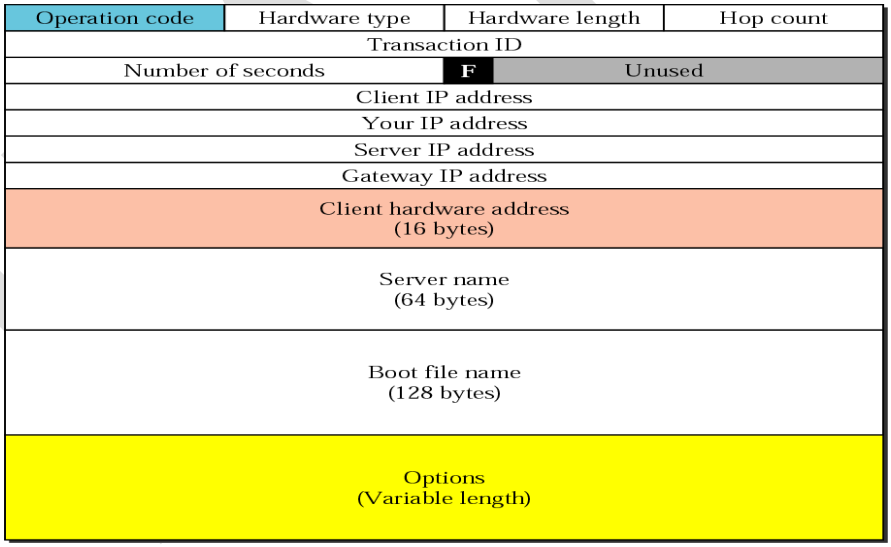
Packet Format

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. Other options define parameters such as lease time and so on. The options field in DHCP can be up to 312 bytes.



Client IP address. This is a 4-byte field that contains the client IP address. If the client does not have this information, this field has a value of 0.

Your IP address. This is a 4-byte field that contains the client IP address. It is filled by the server (in the reply message) at the request of the client.

Server IP address. This is a 4-byte field containing the server IP address. It is filled by the server in a reply message.

Gateway IP address. This is a 4-byte field containing the IP address of a router. It is filled by the server in a reply message.

Client hardware address. This is the physical address of the client. Although the server can retrieve this address from the frame sent by the client, it is more efficient if the address is supplied explicitly by the client in the request message.

Server name. This is a 64-byte field that is optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the domain name of the server. If the server does not want to fill this field with data, the server must fill it with all 0s.

Boot filename. This is a 128-byte field that can be optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the full pathname of the boot file. The client can use this path to retrieve other booting information. If the server does not want to fill this field with data, the server must fill it with all 0s.

Options. This is a 64-byte field with a dual purpose. It can carry either additional information (such as the network mask or default router address) or some specific vendor information. The field is used only in a reply message. The server uses a number, called a magic cookie, in the format of an IP address with the value of 99.130.83.99. When the client finishes reading the message, it looks for this magic cookie. If present, the next 60 bytes are options. An option is composed of three fields: a 1-byte tag field, a 1-byte length field, and a variable-length value field. The length field defines the length of the value field, not the whole option.

Address Discovery and Binding

The DHCP has been devised to provide static and dynamic address allocation.

Static Address Allocation

In this capacity, a DHCP server has a database that statically binds physical addresses to IP addresses. When working in this way, DHCP is backward compatible with the deprecated protocol BOOTP, which we discussed before.

Dynamic Address Allocation

DHCP has a second database with a 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 (unused) 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 (for example, a subscriber to a service provider). DHCP provides temporary IP addresses 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. Transition States To provide dynamic address allocation, the DHCP client acts as a state machine that performs transitions from one state to another depending on the messages it receives or sends. The type of the message in this case is defined by the option with tag 53 that is included in the DHCP packet. In other words, instead of adding one extra field to the BOOTP protocol to define DHCP type, the designer decided to add an extra option for this purpose. Figure 18.7 shows the type option and the interpretation of its value to define the type of the DHCP packet.

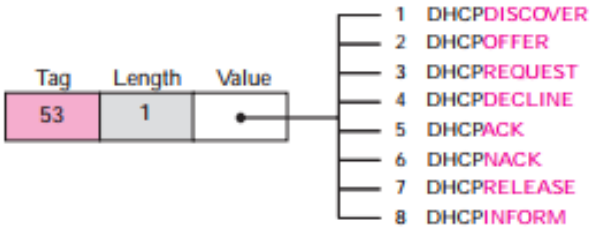
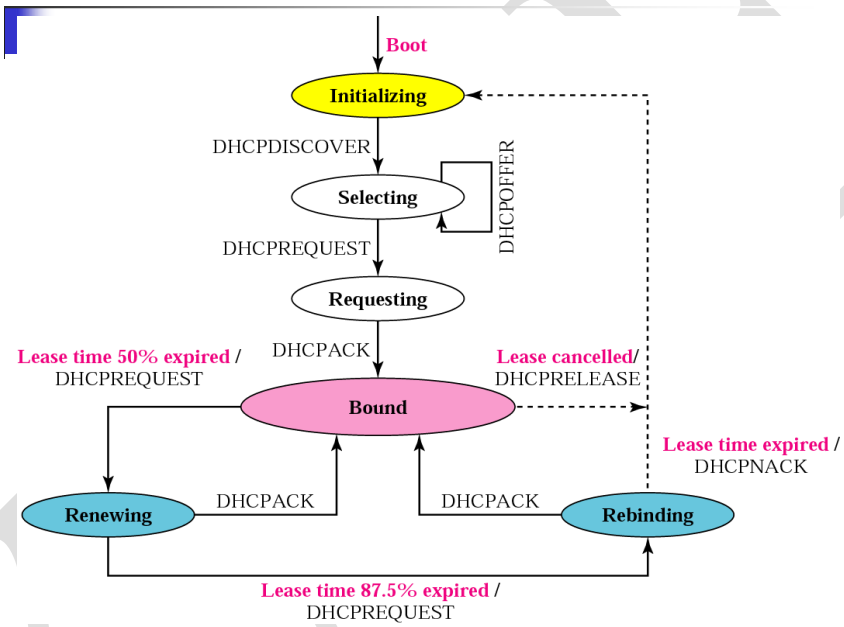


Figure shows the transition diagram with main states.



INIT State

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

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. They can also offer the lease duration. The default is 1 hour. 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 the 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 2 seconds. If there is no reply to any of these DHCPDISCOVERs, the client sleeps for 5 minutes before trying again.

REQUESTING State

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

BOUND State

In this state, the client can use the IP address until the lease expires. When 50 percent of the lease period is reached, the client sends another DHCPREQUEST to ask for renewal. It then goes to the renewing state. When in the bound state, the client can also cancel the lease and go to the initializing state.

RENEWING State

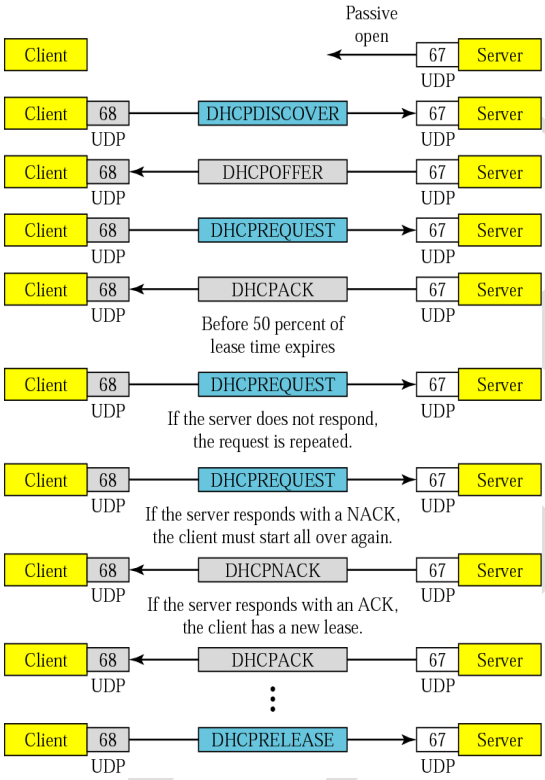
The client remains in the renewing state until one of two events happens. It can receive a DHCPACK, which renews the lease agreement. In this case, the client resets its timer and goes back to the bound state. Or, if a DHCPACK is not received, and 87.5 percent of the lease time expires, the client goes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

REBINDING State

The client remains in the rebinding state until one of three events happens. If the client receives a DHCPNACK 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 shows the exchange of messages related to the transition diagram.



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 he 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 is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

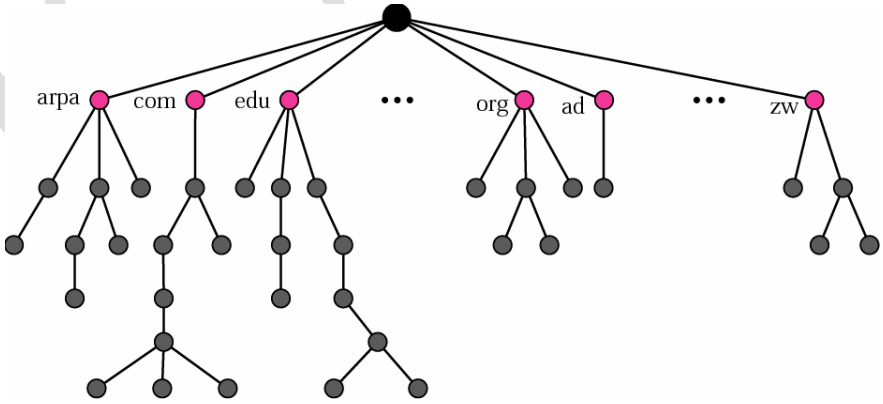
Hierarchical Name Space

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, 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).

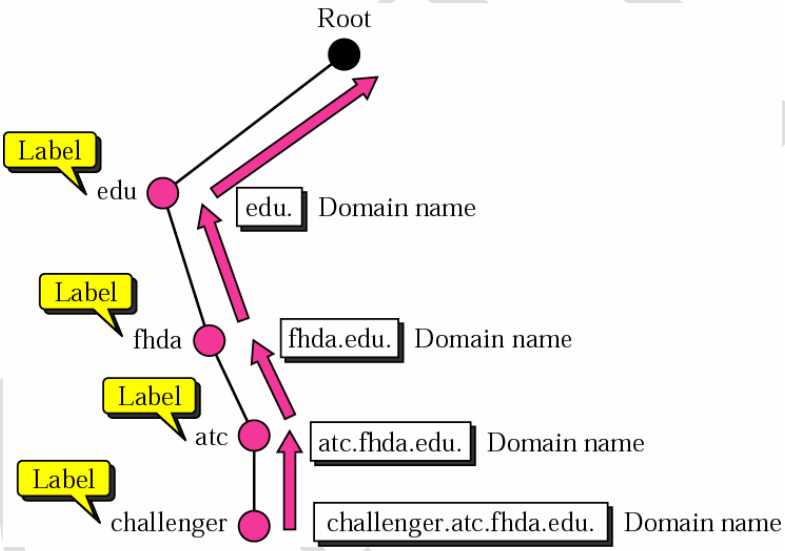


Label

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.

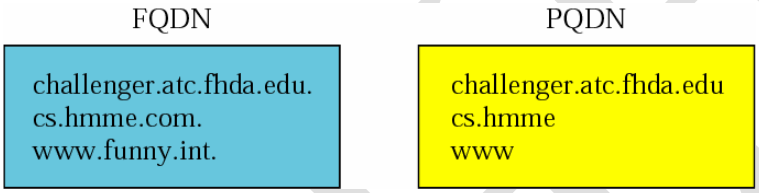


Fully Qualified Domain Name (FQDN)

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

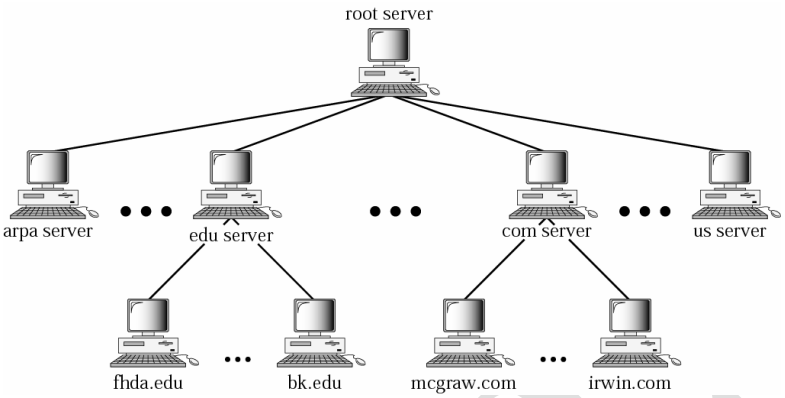


Domain

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

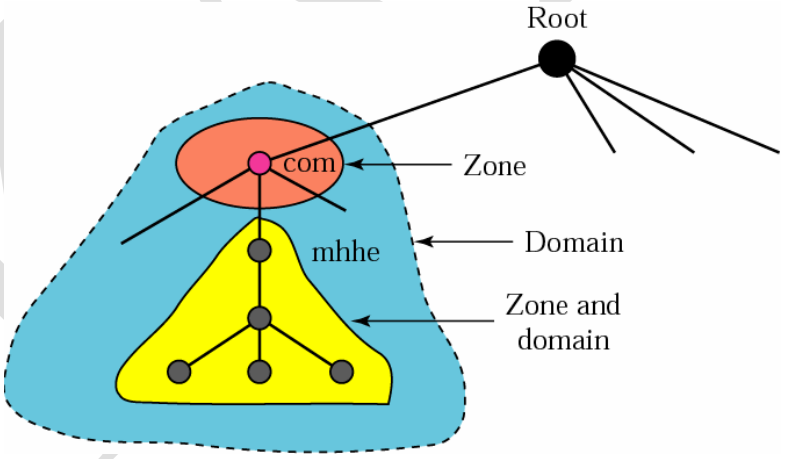
Hierarchy of Name Servers

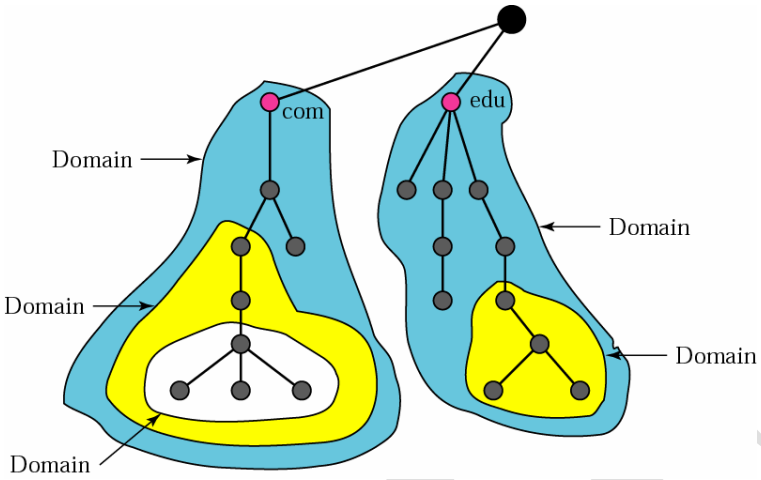
The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level. In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level 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 subdomains and delegates part of its





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.

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

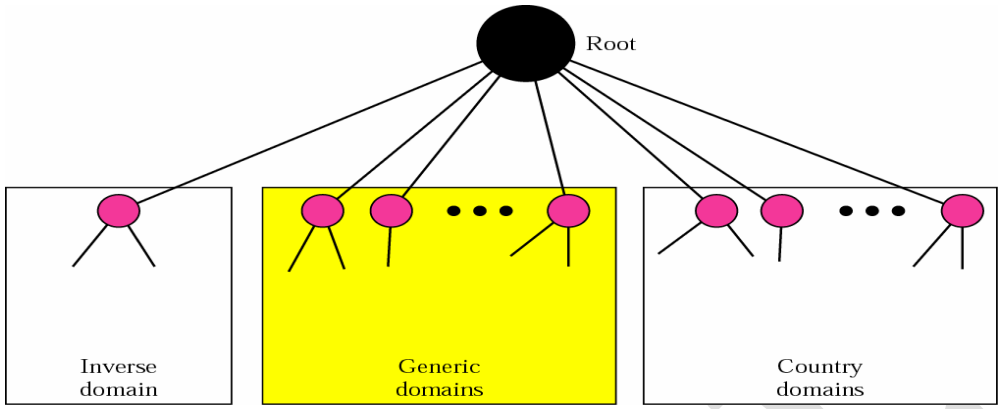
DNS defines two types of servers: primary and secondary. A primary server is a server that stores a file about the zone for which it is an authority. It is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.

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.

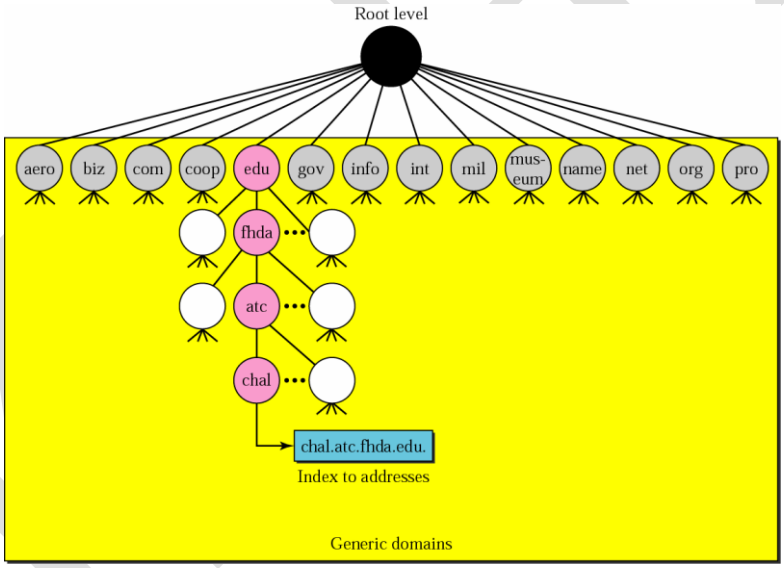
DNS in Internet

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.



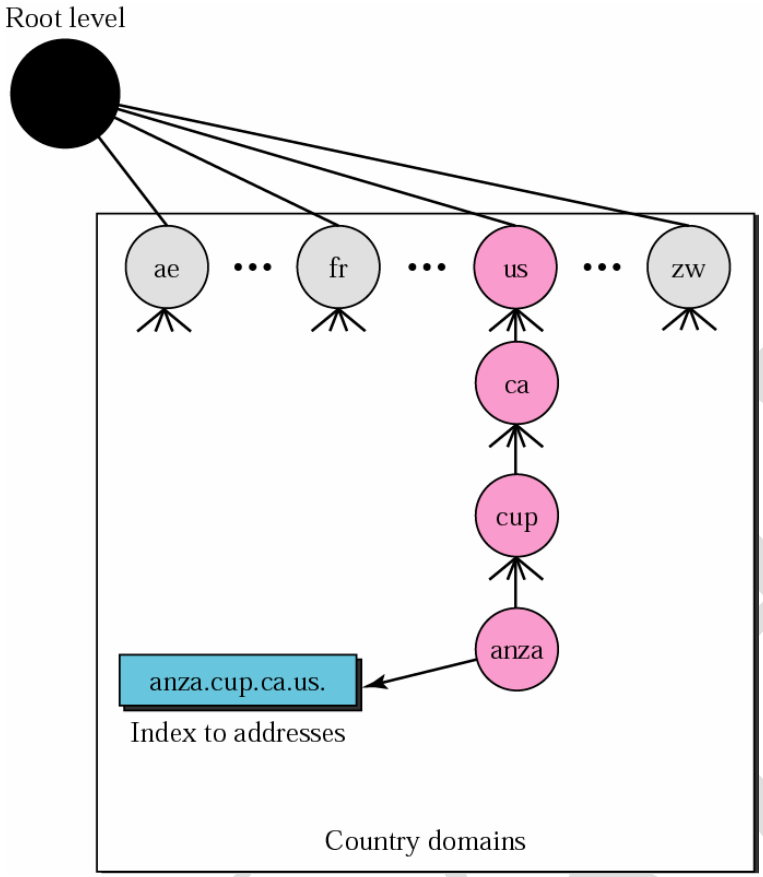
Looking at the tree, we see that the first level in the generic domains section allows 14 possible labels. These labels describe the organization types as listed in table.

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.

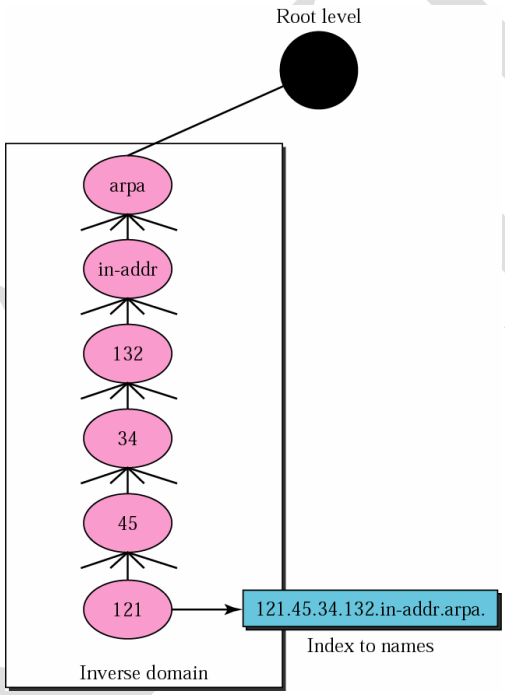


Inverse domain

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 IP packet) 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.

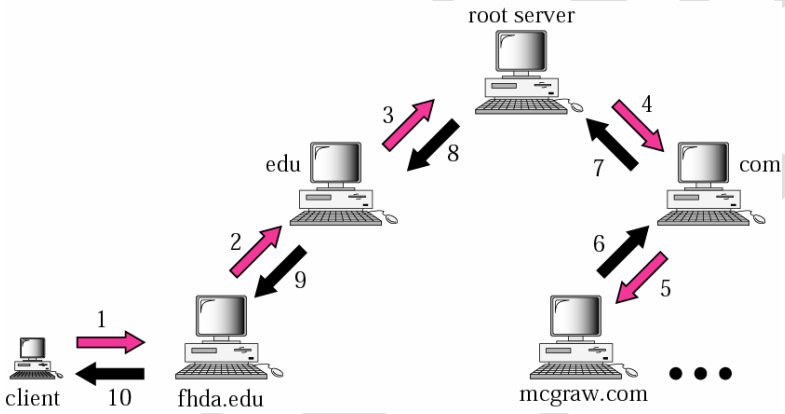
Mapping Addresses to Names

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.



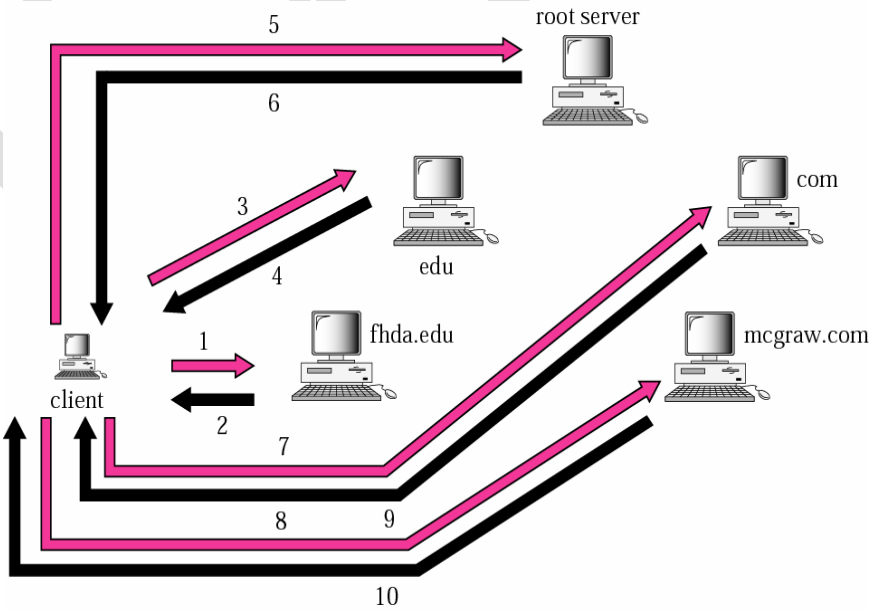
Iterative Resolution

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 unauthoritative.

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.

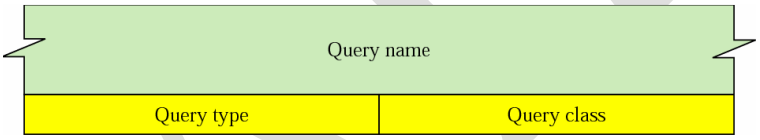


Resource Records

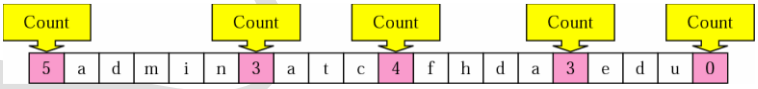
Two types of records are used in DNS. The question records are used in the question section of the query and response messages. The resource records are used in the answer, authoritative and additional information sections of the response message.

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).

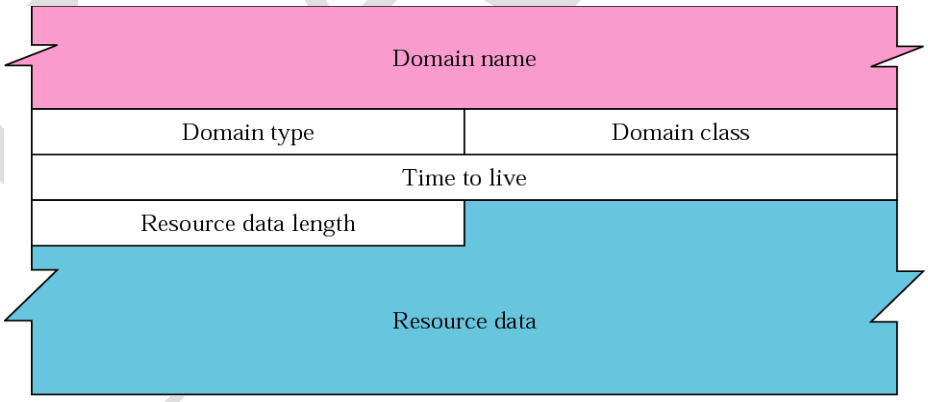


Query type. This is a 16-bit field defining the type of query. Table below shows some of the types commonly used. The last two can only be used in a query.

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 MARKS

1. Define BOOTP.
2. What is the purpose of DHCP?
3. Mention the fields in DHCP.
4. List the difference between static address and dynamic addressing.
5. Define DNS.
6. What is namespace?
7. How do you map names to addresses?
8. What are generic domains?
9. Differentiate FQDN and PQDN.
10. What are resource records?

EIGHT MARKS

1. Explain the operation of BOOTP.
2. Discuss the message format of BOOTP.
3. Explain the role of DHCP.
4. Write in brief about transition states of DHCP.
5. Discuss about name space and domain name space with example.
6. Write about the function of DNS in internet.
7. Brief about resolution
8. Discuss about mapping names to address and address to names.
9. Brief about types of records used in DNS.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

ONE MARK QUESTIONS

DEPARTMENT OF COMPUTER SCIENCE

STAFF NAME: S.MANJU PRIYA **SUBJECT NAME: INTERNETWORKING WITH TCP/IP**

SUB.CODE: 17CSP201 **UNIT IV** **SEMESTER: II**

S.NO	Question	Choice1	Choice2	Choice3	Choice4	Ans
1	_____is a client/server protocol designed to provide information for a disk less computer	BOOTP	RARP	ICMP	ARP	BOOTP
2	Bootstrap protocol is an _____ layer program.	network	transport	application	physical	application
3	The host that can be used as a relay to operate at the application layer is _____	BOOTP server	BOOT client	relay agent	request agent	relay agent
4	The relay agent knows the _____address of a BOOT server	multicast	unicast	broadcast	network address	unicast
5	The ID which is randomly choosen for each connection involving BOOTP is_____	transaction ID	net ID	host id	BOOTP id	transaction ID

6	BOOTP uses _____which does not provide error control	TCP	UDP	TFTP	ARP	UDP
7	_____ 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
8	_____ is a 4 bit field containing the IP address of a router	gate way IP address	client IP address	server IP address	router ip	gate way IP address
9	_____ provides static and dynamic address allocation that can be manual or automatic.	BOOTP	DHCP	TFTP	RARP	DHCP
10	_____ is backward compatible with BOOTP	DHCP	BOOTP	UDP	ARP	DHCP
11	DHCP server issues a_____for a specific period of time	time stamp	time slice	lease	time elapsed	lease
12	A server reply can be _____	broadcast	unicast	multicast	nocast	both a&b
13	_____is a static configuration protocol	BOOTP	TFPT	UDP	RARP	BOOTP
14	_____ is a client server application on the internet with the unique user friendly name	DNS	DDNS	FQDN	PQDN	DNS
15	A _____ server gets its information from a primary server	root server	DNS server	BOOTP server	secondary server	secondary server
16	_____ server creates maintains and updates information about its zone	secondary	primary	root	BOOTP	primary
17	The domain name space is divided into _____ sections.	2	4	3	7	3

18	The DNS client alled a _____maps a name to a address or an address to a name	resolver	register	server	client	resolver
19	IN _____ resolution the client sends its request to a server	Iterative	recursive	non recursive	non iterative	recursive
20	In _____resolution the client may send its request to multiple servers	iterative	recursive	non recursive	non iterative	iterative
21	Two types of DNS messages are _____	questions and resources	Question and answers	queries and response	resources and records	queries and response
22	_____ is a method where by an answer to query is stored in memory	queuing	stacking	querying	caching	caching
23	DNS uses an __pointer for duplicated domain name information in its message	offset	inset	static	dynamic	offset
24	A_____ maps each address to a unique name	address space	domain space	offset pointer	name space	name space
25	_____ is a sequence of characters without structures	hierarchal name space	flat name space	address name space	domain name space	flat name space
26	In _____ name space the names are defined in a inverted tree structure	hierarchical	flat	address	domain	domain
27	_____ is a sting with the maximum of 63 characters	label	domain name	FQDN	none	label
28	The tree can have only _____levels	126	127	128	138	128
29	_____ is a sequence of labels separated by dots	domain name	FQDN	address space	PQDN	domain name

30	_____ is a domain name that contains the full name of a host	PQDN	FQDN	QQDN	AQDN	FQDN
31	A _____ is sub tree of the domain name space	zone	PQDN	domain	FQDN	PQDN
32	____ is used when the name to be resolved belongs to the same site as client	PQDN	FQDN	address space	address resloution	FQDN
33	What a server is responsible for or has authority over is called a _____	segment	zone	domain	packed	zone
34	A _____is a server whose zone consists of the whole tree	primary server	secondary server	com server	root server	root server
35	The _____ define registered hosts according to their generic behavior	country domain	generic domain	inverse domain	net domain	generic domain
36	The _____section uses two character country abbreviations	country domain	generic domain	country domain	net domain	country domain
37	_____ domain is used to map an address to a name	inverse	generic	country	net	inverse
38	The new domains are added to the DNS by a _____	resolver	registrar	server	client	registrar
39	_____record is used by the client to get information from a server	question	answer	query	authoritative	question
40	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
41	BOOTP stands for	Bootstrap protocol	Bootstrap project	Booting protocol	Booting project	Bootstrap protocol

42	What is the default behavior of R1 when PC1 requests service from DHCP server?	Drop the request	Broadcast the request to R2 and R3	Forward the request to R2	Broadcast the request to R2, R3 and ISP	Drop the request
43	_____. Is the process of placing timestamps on dynamically registered records	Aging	IP	AIPAA	TCP	Aging
44	DHCP ____ _____ occur every 60 minutes	Multicast Scope	Super Scope	Subnet Mask	Automatic Backups	Automatic Backups
45	_____ interval is the period after the timestamp is set that must elapse before refresh can occur.	Nonsecure Method	Refresh Interval	Forwarder	No-refresh Interval	No-refresh Interval
46	internetworking protocol for routing packets over a network is called_____.	IP	AIPAA	TCP	Aging	IP
47	Name of domain is domain name of node at top of the	Sub Tree	Main Tree	Leaf Node	Bottom Tree	Sub Tree
48	Domain, which is used to map an address to a name is called	Generic Domains	Inverse Domain	Main Domains	Sub-Domain	Inverse Domain
49	Well-known port used for encapsulation by server is	Port 7	Port 23	Port 53	Port 67	Port 53
50	Each node in tree has a	Primary Name	Domain name.	DNS tree	host tree	Domain name.
51	Domain Name System (DNS), is a protocol that can be used in different	Layers	Categories	Platform	Stages	Platform
52	In Internet, domain name tree is divided into three	Different Stages	Different Layers	Different Parts	Different Sections	Different Sections

53	New domains can be added to DNS through a	Query	Registrar	Domain	Response	Registrar
54	A supporting program that is used by other programs such as e-mail is called	DNS	SMTP	IP	Server/Client	DNS
55	Country domains section uses two-character country	Generations	Abbreviations	Notations	Zones	Abbreviations
56	In Domain Name System (DNS), a contiguous part of entire tree is called	Host	Server	Domain	Zone	Zone

UNIT-V

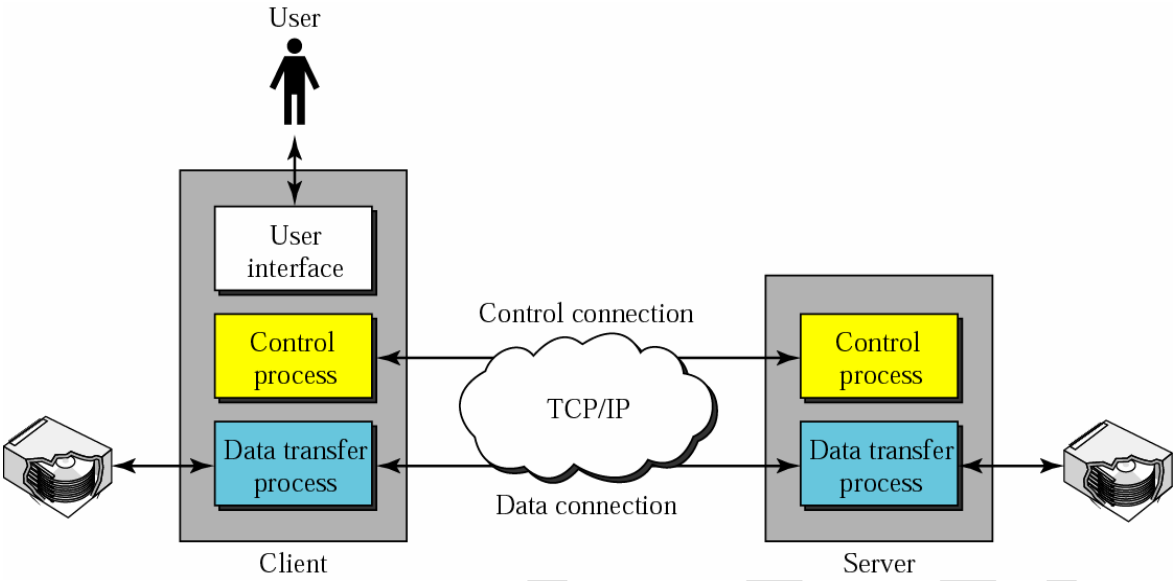
SYLLABUS

Remote Login : FTP – SMTP – SNMP. IP over ATM -ATMWAN– Routing the Cells – ATMARP – Logical IP Subnets. VPN

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 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

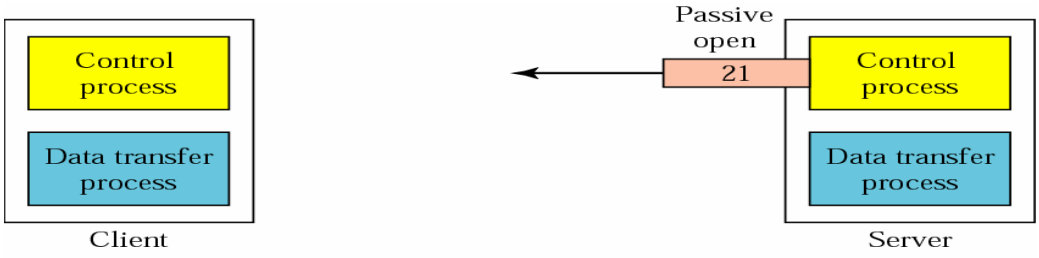
The two FTP connections control and data use different strategies and different port numbers.

Control Connection

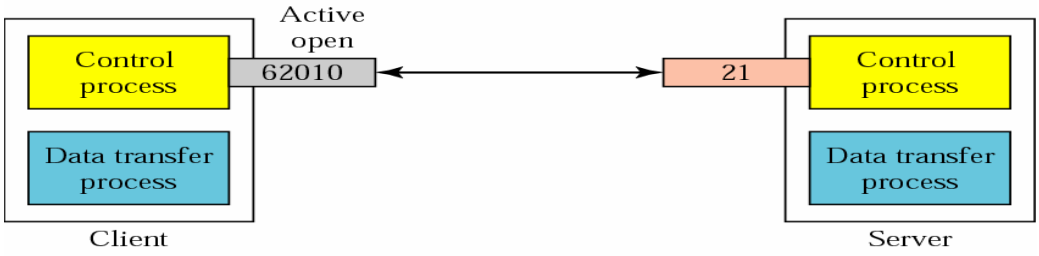
The control connection is created in the same way as other application programs described so far. There are two steps:

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 *minimize 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 shows the initial connection between the server and the client.



a. Passive open by server



b. Active open by client

Data Connection

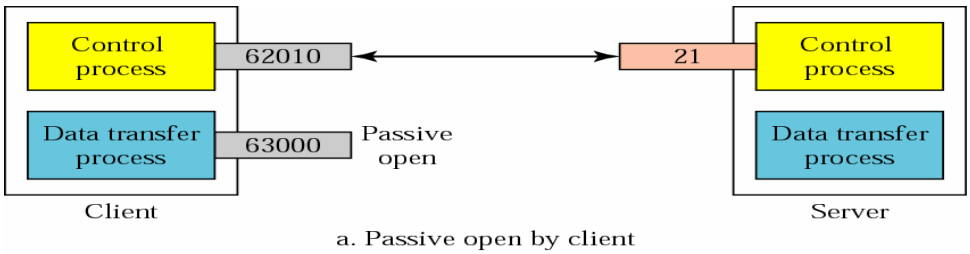
The **data connection** uses the well-known port 20 at the server site. However, the creation of a data connection is different from what we have seen so far. The following shows how FTP creates a data connection:

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.

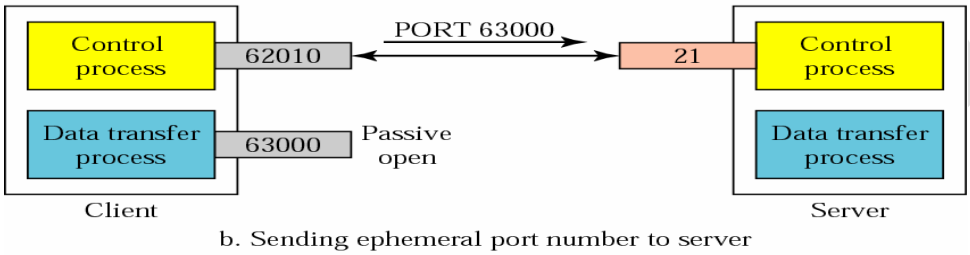
Communication

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.

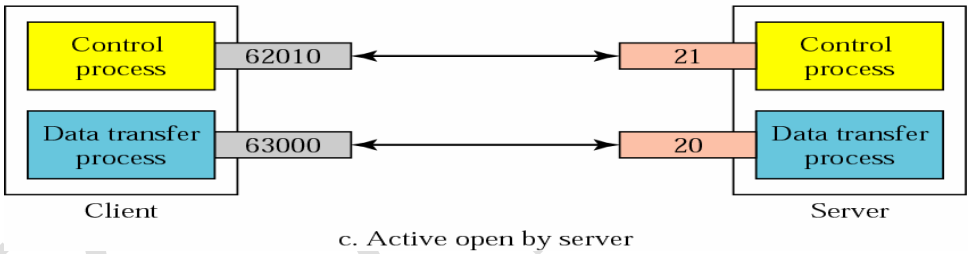
` FTP has two different approaches, one for the control connection and one for the data communication. We will study each approach separately.



a. Passive open by client



b. Sending ephemeral port number to server



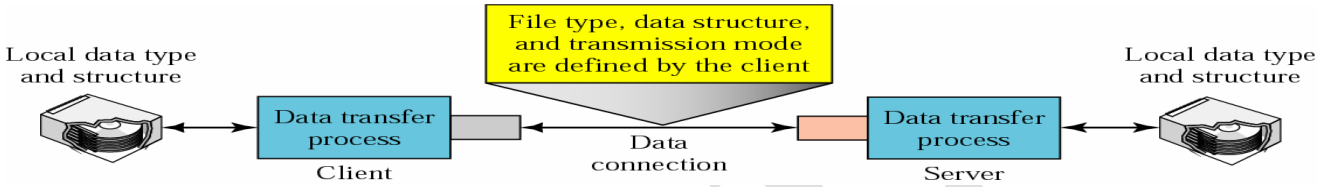
c. Active open by server

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

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).



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.

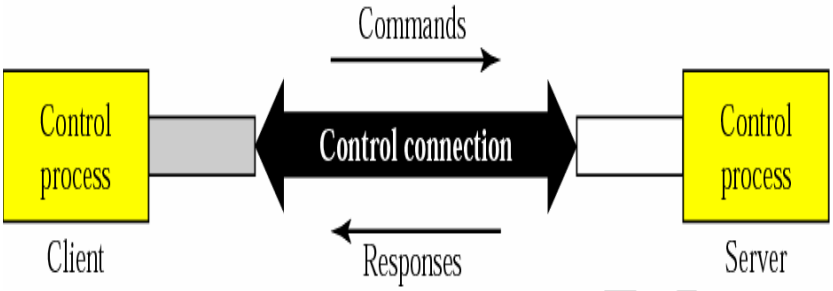
Transmission mode FTP can transfer a file across the data connection using one of the following three transmission modes:

- **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).



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 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). 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 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

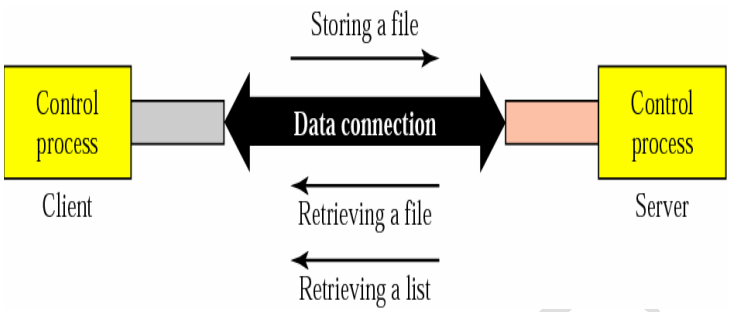
125	Data connection open; data transfer will start shortly
150	File status is OK; data connection will be open shortly
Positive completion reply	
200	Command OK
211	System status or help reply
212	Directory status
213	File status
214	Help message
215	Naming the system type(operating system)
220	Service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

- 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.



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

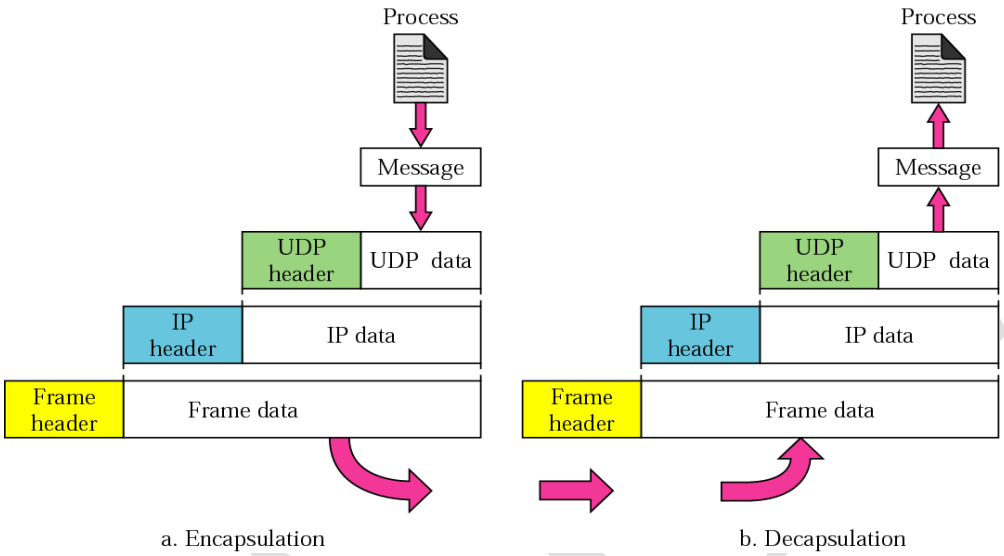
UDP is a very simple, unreliable transport protocol. There is no flow control, and hence no window mechanism. The receiver may overflow with incoming messages.There is no error control mechanism in UDP except for the checksum. This means thus the sender does not know if message has been lost are duplicated. 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

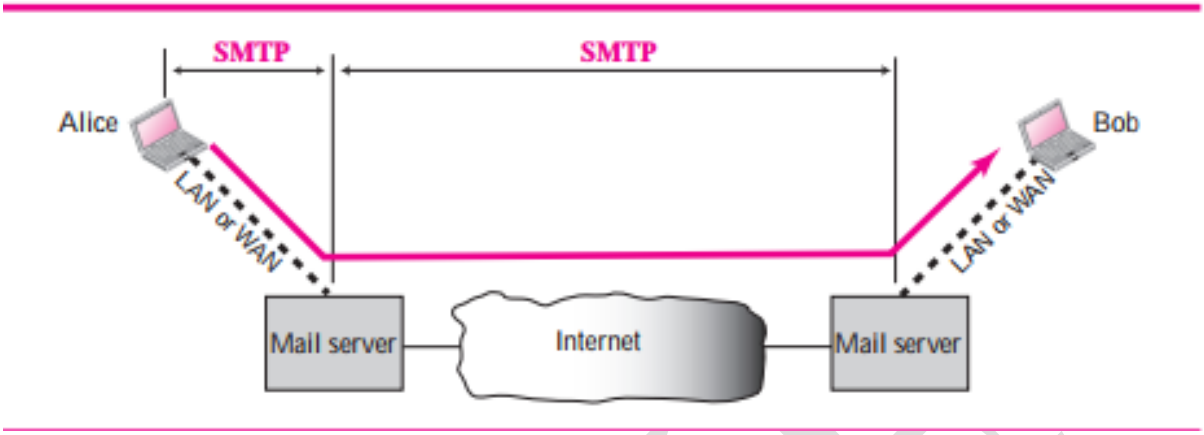
Encapsulation

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.



SMTP- Simple Mail Transfer Protocol

The actual mail transfer is done through message transfer agents (MTAs). To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP). As we said before, two pairs of MTA clientserver programs are used in the most common situation (fourth scenario). Figure shows the range of the SMTP protocol in this scenario.



SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. As we will see shortly, another protocol is needed between the mail server and the receiver. SMTP simply defines how commands and responses must be sent back and forth. Each network is free to choose a software package for implementation.

Commands and Responses

SMTP uses commands and responses to transfer messages between an MTA client and an MTA server (see Figure).

Each command or reply is terminated by a two-character (carriage return and line feed) end-of-line token.

Commands

Commands are sent from the client to the server. The format of a command is shown below: It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands listed in Table and described in more detail below.

Keyword	*Argument(s)*	*Keyword*	*Argument(s)*
HELO	Sender's host name	NOOP	
MAIL FROM	Sender of the message	TURN	
RCPT TO	Intended recipient	EXPN	Mailing list
DATA	Body of the mail	HELP	Command name
QUIT		SEND FROM	Intended recipient
RSET		SMOL FROM	Intended recipient
VRFY	Name of recipient	SMAL FROM	Intended recipient

HELO. This command is used by the client to identify itself. The argument is the domain name of the client host. The format is

HELO: challenger.atc.fhda.edu

MAIL FROM. This command is used by the client to identify the sender of the message. The argument is the e-mail address of the sender (local part plus the domain name). The format is

MAIL FROM: forouzan@challenger.atc.fhda.edu

RCPT TO. This command is used by the client to identify the intended recipient of the message. The argument is the e-mail address of the recipient. If there are multiple recipients, the command is repeated. The format is

RCPT TO: betsy@mcgraw-hill.com

DATA. This command is used to send the actual message. All lines that follow the DATA command are treated as the mail message. The message is terminated by a line containing just one period. The format is

DATA

This is the message to be

sent to the McGraw-Hill Company.

.

QUIT. This command terminates the message. The format is

QUIT

RSET. This command aborts the current mail transaction. The stored information about the sender and recipient is deleted. The connection will be reset.

RSET

VRFY. This command is used to verify the address of the recipient, which is sent as the argument. The sender can ask the receiver to confirm that a name identifies a valid recipient. Its format is

VRFY: betsy@mcgraw-hill.com

NOOP. This command is used by the client to check the status of the recipient. It requires an answer from the recipient. Its format is

NOOP

TURN. This command lets the sender and the recipient switch positions, whereby the sender becomes the recipient and vice versa. However, most SMTP implementations today do not support this feature. The format is

TURN

EXPN. This command asks the receiving host to expand the mailing list sent as the arguments and to return the mailbox addresses of the recipients that comprise the list. The format is

HELP. This command asks the recipient to send information about the command sent as the argument. The format is

HELP: mail

SEND FROM. This command specifies that the mail is to be delivered to the terminal of the recipient, and not the mailbox. If the recipient is not logged in, the mail is bounced back. The argument is the address of the sender. The format is

SMOL FROM: forouzan@fhda.atc.edu

SMOL FROM. This command specifies that the mail is to be delivered to the terminal and the mailbox of the recipient. This means that if the recipient is logged in, the mail is delivered to the terminal and the mailbox. If the recipient is not logged in, the mail is delivered only to the mailbox. The argument is the address of the sender. The format is

SMAL FROM: forouzan@fhda.atc.edu

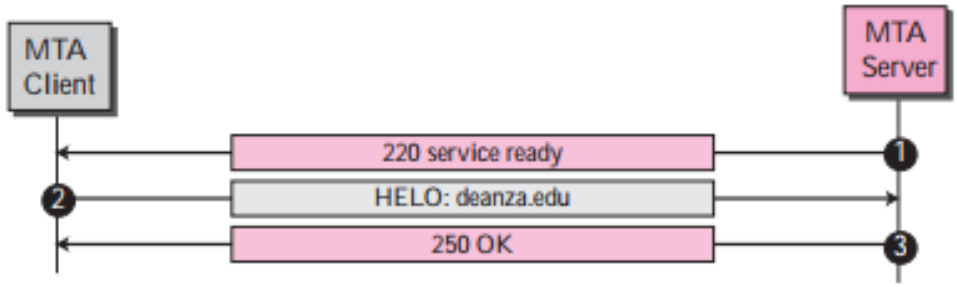
Responses

Responses are sent from the server to the client. A response is a three-digit code that may be followed by additional textual information. Table lists some of the responses.

Code	*Description*
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination. Connection Establishment After a client has made a TCP connection to the well-known port 25, the SMTP server starts the connection phase. This phase involves the following three steps, which are illustrated in Figure .



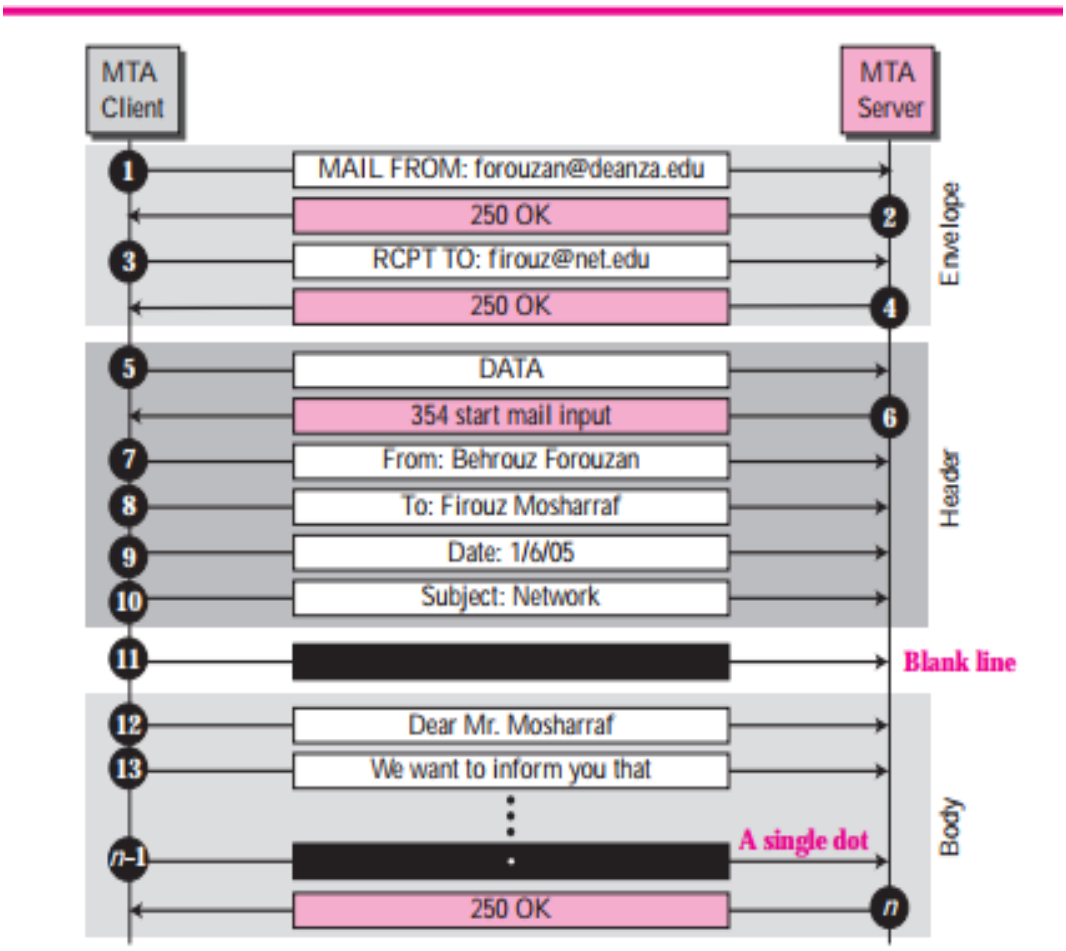
The server sends code 220 (service ready) to tell the client that it is ready to receive mail. If the server is not ready, it sends code 421 (service not available).

2. The client sends the HELO message to identify itself using its domain name address. This step is necessary to inform the server of the domain name of the client. Remember that during TCP connection establishment, the sender and receiver know each other through their IP addresses.

3. The server responds with code 250 (request command completed) or some other code depending on the situation.

Message Transfer

After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged. This phase involves eight steps. Steps 3 and 4 are repeated if there is more than one recipient (see Figure).

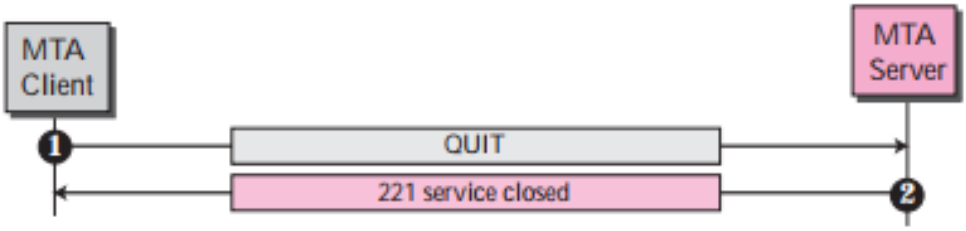


1. The client sends the MAIL FROM message to introduce the sender of the message. It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages
2. . 2. The server responds with code 250 or some other appropriate code.
3. 3. The client sends the RCPT TO (recipient) message, which includes the mail address of the recipient.
4. 4. The server responds with code 250 or some other appropriate code.
5. 5. The client sends the DATA message to initialize the message transfer.
6. The server responds with code 354 (start mail input) or some other appropriate message.

7. 7. The client sends the contents of the message in consecutive lines. Each line is terminated by a two-character end-of-line token (carriage return and line feed). The message is terminated by a line containing just one period.
8. 8. The server responds with code 250 (OK) or some other appropriate code.

Connection Termination

After the message is transferred successfully, the client terminates the connection. This phase involves two steps (see Figure).



1. The client sends the QUIT command.
2. The server responds with code 221 or some other appropriate code.

After the connection termination phase, the TCP connection must be closed.

SNMP

What is SNMP?

Simple Network Management Protocol (SNMP) is an application–layer protocol defined by the Internet Architecture Board (IAB) in RFC1157 for exchanging management information between network devices. It is a part of Transmission Control ProtocoИnternet Protocol (TCPИP) protocol suite. SNMP is one of the widely accepted protocols to manage and monitor network elements. Most of the professional–grade network elements come with bundled SNMP agent. These agents have to be enabled and configured to communicate with the network management system (NMS).

SNMP basic components and their functionalities

SNMP consists of SNMP Manager, Managed devices, SNMP agent Management Information Database Otherwise called as Management Information Base (MIB).

SNMP Manager:

A manager or management system is a separate entity that is responsible to communicate with the SNMP agent implemented network devices. This is typically a computer that is used to run one or more network management systems.

SNMP Manager's key functions

- Queries agents
- Gets responses from agents
- Sets variables in agents
- Acknowledges asynchronous events from agents
-

Managed Devices:

A managed device or the network element is a part of the network that requires some form of monitoring and management e.g. routers, switches, servers, workstations, printers, UPSs, etc...

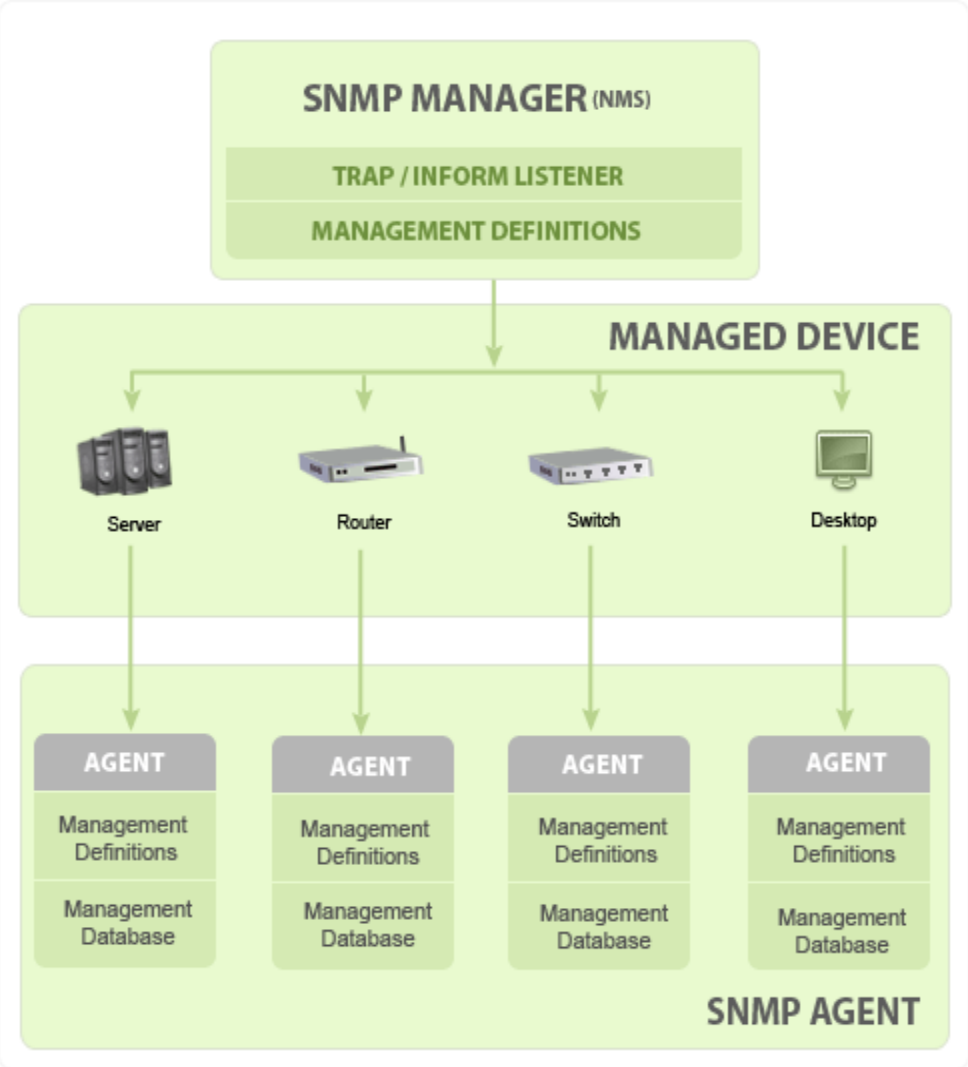
SNMP Agent:

The agent is a program that is packaged within the network element. Enabling the agent allows it to collect the management information database from the device locally and makes it available to the SNMP manager, when it is queried for. These agents could be standard (e.g. Net-SNMP) or specific to a vendor (e.g. HP insight agent)

SNMP agent's key functions

- Collects management information about its local environment
- Stores and retrieves management information as defined in the MIB.
- Signals an event to the manager.
- Acts as a proxy for some non–SNMP manageable network node.

Basic SNMP Communication Diagram



Management Information database or Management Information Base (MIB)

Every SNMP agent maintains an information database describing the managed device parameters. The SNMP manager uses this database to request the agent for specific information and further translates the information as needed for the Network Management System (NMS).

This commonly shared database between the Agent and the Manager is called Management Information Base (MIB).

Typically these MIB contains standard set of statistical and control values defined for hardware nodes on a network. SNMP also allows the extension of these standard values with values specific to a particular agent through the use of private MIBs.

In short, MIB files are the set of questions that a SNMP Manager can ask the agent. Agent collects these data locally and stores it, as defined in the MIB. So, the SNMP Manager should be aware of these standard and private questions for every type of agent.

MIB structure and Object Identifier (Object ID or OID)

Management Information Base (MIB) is a collection of Information for managing network element. The MIBs comprises of managed objects identified by the name Object Identifier (Object ID or OID).

Each Identifier is unique and denotes specific characteristics of a managed device. When queried for, the return value of each identifier could be different e.g. Text, Number, Counter, etc...

There are two types of Managed Object or Object ID: Scalar and Tabular. They could be better understandable with an example

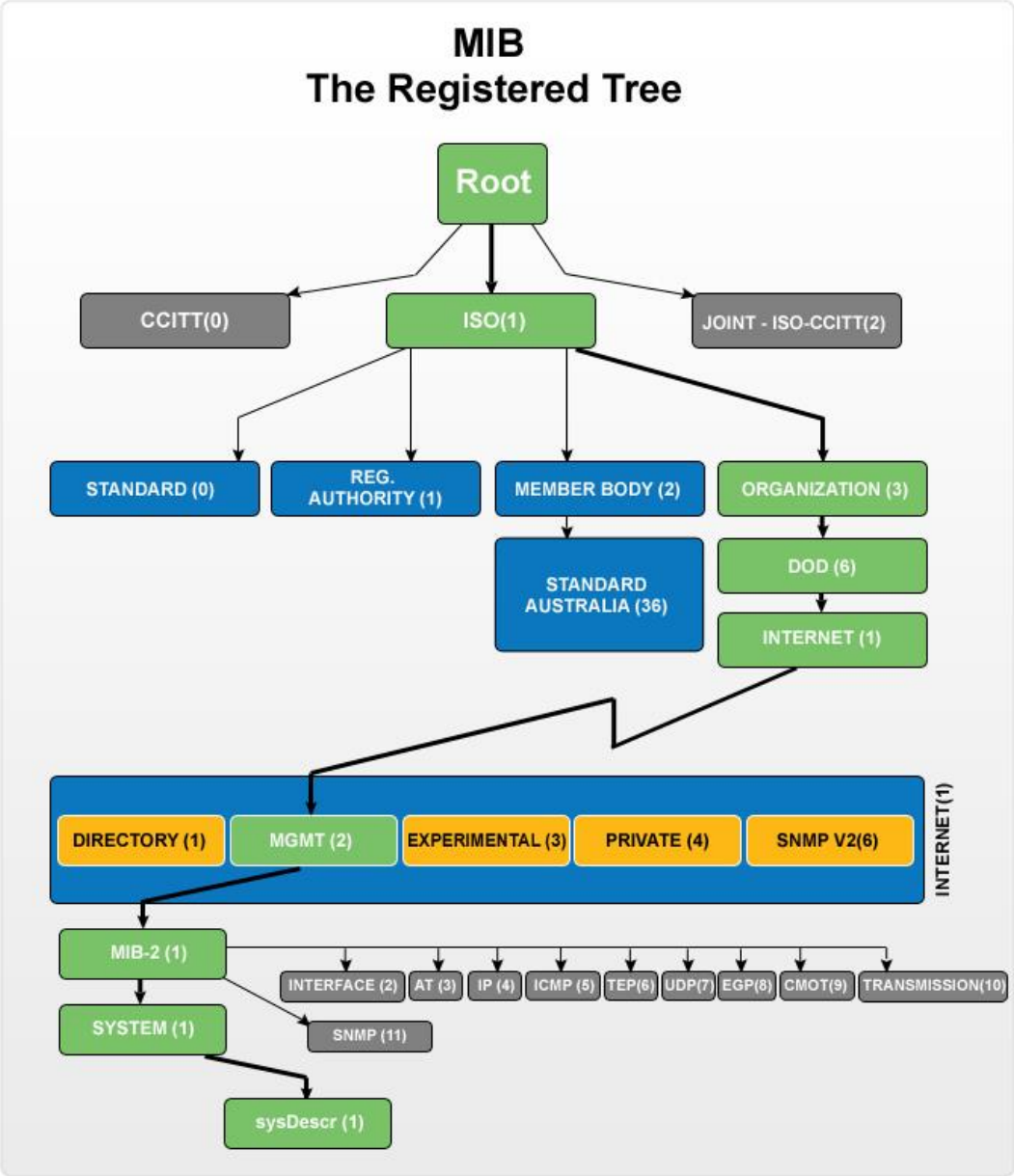
Scalar: Device's vendor name, the result can be only one. (As definition says: "Scalar Object define a single object instance")

Tabular: CPU utilization of a Quad Processor, this would give me a result for each CPU separately, means there will be 4 results for that particular Object ID. (As definition says: "Tabular object defines multiple related object instance that are grouped together in MIB tables")

Every Object ID is organized hierarchically in MIB. The MIB hierarchy can be represented in a tree structure with individual variable identifier.

A typical object ID will be a dotted list of integers. For example, the OID in RFC1213 for "sysDescr" is .1.3.6.1.2.1.1.1

MIB Tree Diagram



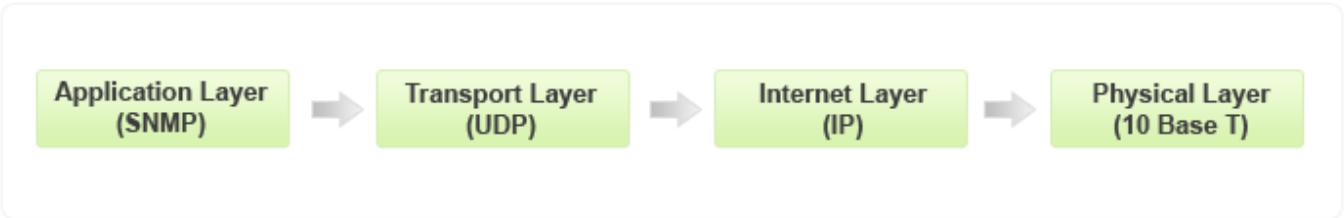
Basic commands of SNMP

The simplicity in information exchange has made the SNMP as widely accepted protocol. The main reason being concise set of commands, here are they listed below:

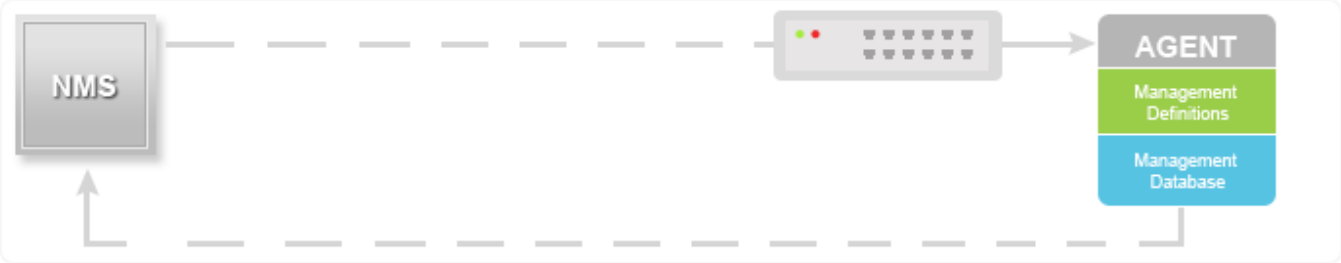
- GET: The GET operation is a request sent by the manager to the managed device. It is performed to retrieve one or more values from the managed device.
- GET NEXT: This operation is similar to the GET. The significant difference is that the GET NEXT operation retrieves the value of the next OID in the MIB tree.
- GET BULK: The GETBULK operation is used to retrieve voluminous data from large MIB table.
- SET: This operation is used by the managers to modify or assign the value of the Managed device.
- TRAPS: Unlike the above commands which are initiated from the SNMP Manager, TRAPS are initiated by the Agents. It is a signal to the SNMP Manager by the Agent on the occurrence of an event.
- INFORM: This command is similar to the TRAP initiated by the Agent, additionally INFORM includes confirmation from the SNMP manager on receiving the message.
- RESPONSE: It is the command used to carry back the value(s) or signal of actions directed by the SNMP Manager.

Typical SNMP communication

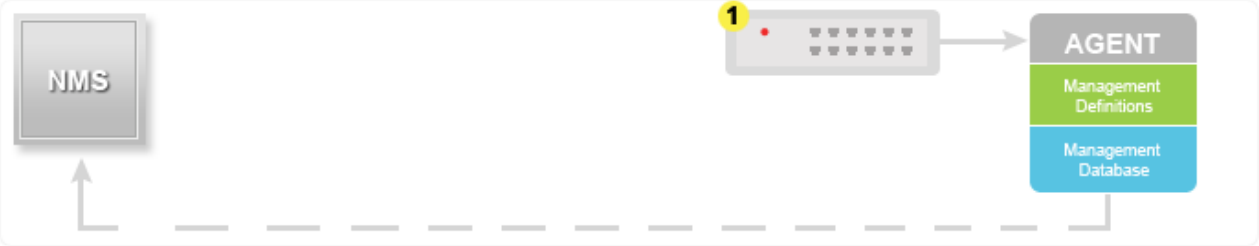
Being the part of TCP/ IP protocol suite, the SNMP messages are wrapped as User Datagram Protocol (UDP) and intern wrapped and transmitted in the Internet Protocol. The following diagram will illustrate the four–layer model developed by Department of Defense (DoD).



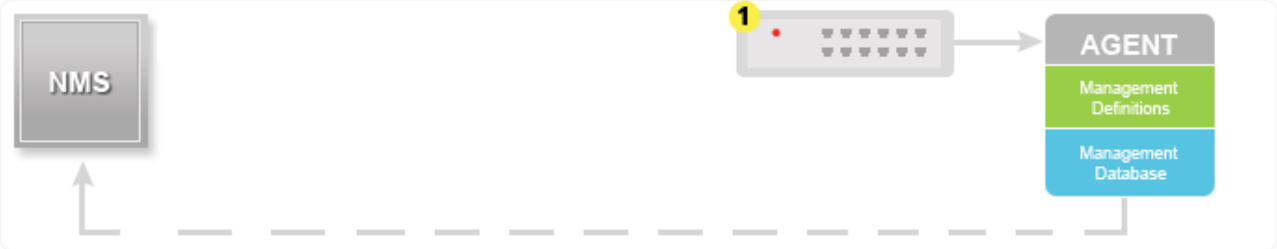
GET/GET NEXT/GET BULK/SET



TRAP



INFORM



By default SNMP uses port 161 and TRAP/INFORM uses port 162 for communication.

SNMP versions

Since the inception SNMP, has gone through significant upgrades. However SNMP v1 and v2c are the most implemented versions of SNMP. Support to SNMP v3 has recently started catching up as it is more secured when compare to its older versions, but still it has not reached considerable market share.

SNMPv1:

This is the first version of the protocol, which is defined in RFCs 1155 and 1157

SNMPv2c:

This is the revised protocol, which includes enhancements of SNMPv1 in the areas of protocol packet types, transport mappings, MIB structure elements but using the existing SNMPv1 administration structure ("community based" and hence SNMPv2c). It is defined in RFC 1901, RFC 1905, RFC 1906, RFC 2578.

SNMPv3:

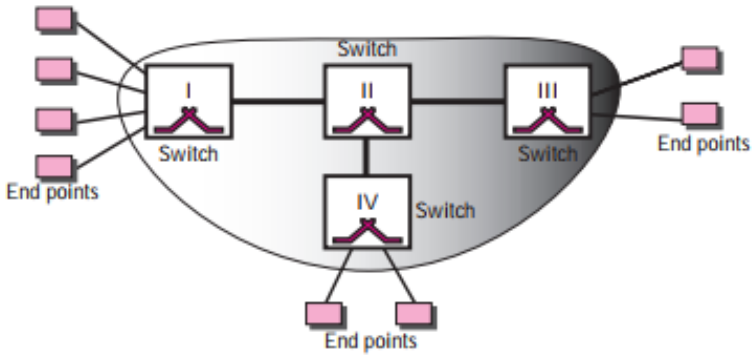
SNMPv3 defines the secure version of the SNMP. SNMPv3 also facilitates remote configuration of the SNMP entities. It is defined by RFC 1905, RFC 1906, RFC 3411, RFC 3412, RFC 3414, RFC 3415.

Though each version had matured towards rich functionalities, additional emphasis was given to the security aspect on each upgrade. Here is a small clip on each editions security aspect.

SNMP v1	Community–based security
SNMP v2c	Community–based security
SNMP v2u	User–based security
SNMP v2	Party–based security
SNMP v3	User–based security

IP over ATM

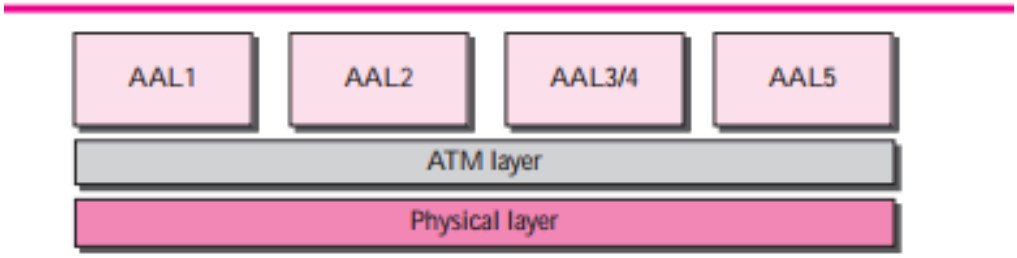
ATM is a switched network. The user access devices, called the end points, are connected to the switches inside the network. The switches are connected to each other using high-speed communication channels. Figure 3.33 shows an example of an ATM network. Virtual Connection Connection between two end points is accomplished through transmission paths (TPs), virtual paths (VPs), and virtual circuits (VCs). A transmission path (TP) is the physical connection (wire, cable, satellite, and so on) between an end point and a switch or between two switches. Think of two switches as two cities. A transmission path is the set of all highways that directly connects the two cities.



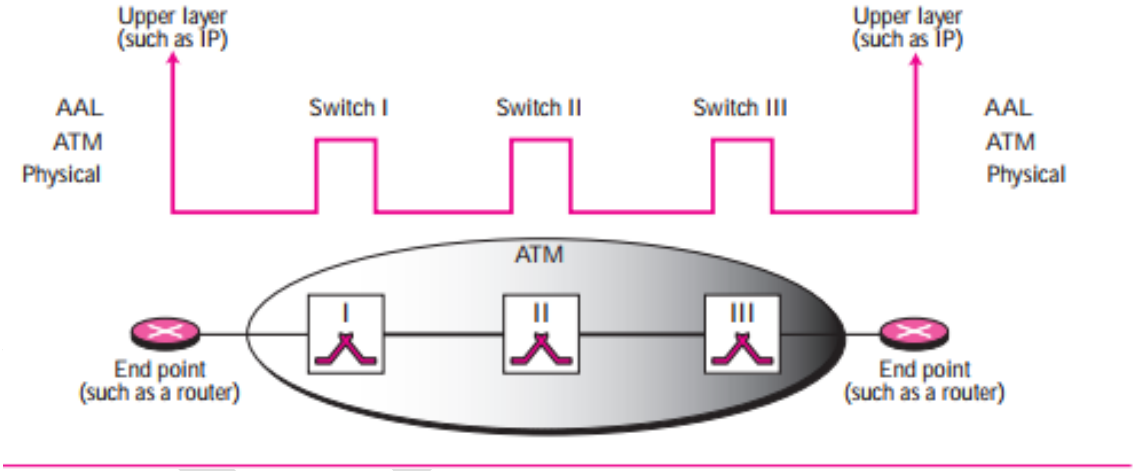
A transmission path is divided into several virtual paths. A virtual path (VP) provides a connection or a set of connections between two switches. Think of a virtual path as a highway that connects two cities. Each highway is a virtual path; the set of all highways is the transmission path. Cell networks are based on virtual circuits (VCs). All cells belonging to a single message follow the same virtual circuit and remain in their original order until they reach their destination.

ATM Layers

The ATM standard defines three layers. They are, from top to bottom, the application adaptation layer, the ATM layer, and the physical layer as shown in Figure.



The physical and ATM layer are used in both switches inside the network and end points (such as routers) that use the services of the ATM. The application adaptation layer (AAL) is used only by the end points. Figure shows the use of these layers inside and outside an ATM network.



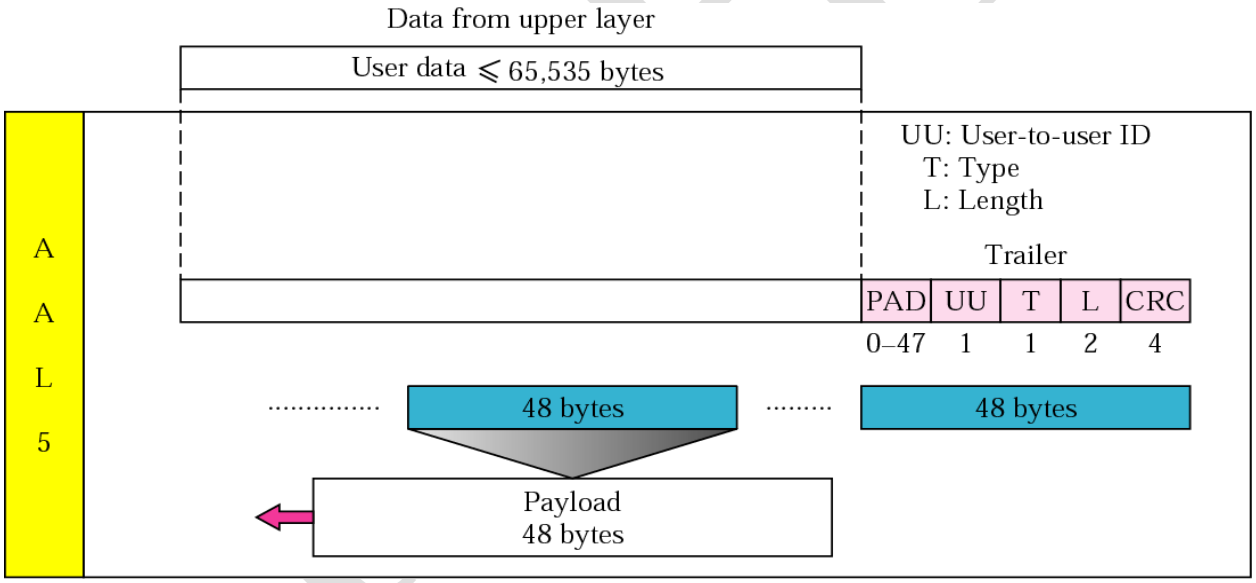
AAL Layer

The application adaptation layer (AAL) allows existing networks (such as packet networks) to connect to ATM facilities. AAL protocols accept transmissions from upper-layer services (e.g., packet data) and map them into fixed-sized ATM cells. These transmissions can be of any type (voice, data, audio, video) and can be of variable or fixed rates. At the receiver, this process is reversed—segments are reassembled into their original formats and passed to the receiving

service. Although four AAL layers have been defined the one which is of interest to us is AAL5, which is used to carry IP packets in the Internet.

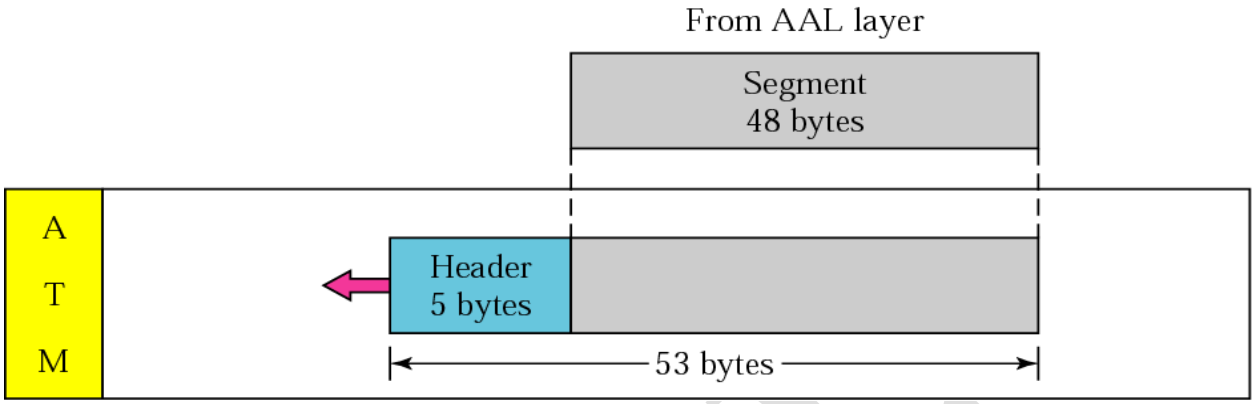
AAL5, which is sometimes called the simple and efficient adaptation layer (SEAL), assumes that all cells belonging to a single message travel sequentially and that control functions are included in the upper layers of the sending application. AAL5 is designed for connectionless packet protocols that use a datagram approach to routing (such as the IP protocol in TCP/IP).

AAL5 accepts an IP packet of no more than 65,535 bytes and adds an 8-byte trailer as well as any padding required to ensure that the position of the trailer falls where the receiving equipment expects it (at the last 8 bytes of the last cell). See Figure. Once the padding and trailer are in place, AAL5 passes the message in 48-byte segments to the ATM layer.

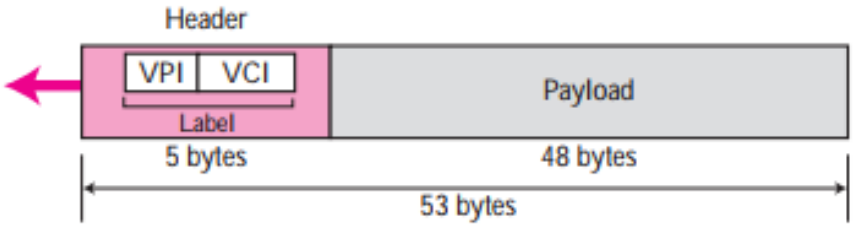


ATM Layer

The ATM layer provides routing, traffic management, switching, and multiplexing services. It processes outgoing traffic by accepting 48-byte segments from the AAL sublayer. The addition of a 5-byte header transforms the segment into a 53-byte cell (see Figure).



A cell is 53 bytes in length with 5 bytes allocated to header and 48 bytes carrying payload (user data may be less than 48 bytes). Most of the header is occupied by the VPI and VCI. Figure shows the cell structure. The combination of VPI and VCI can be thought of as a label that defines a particular virtual connection.

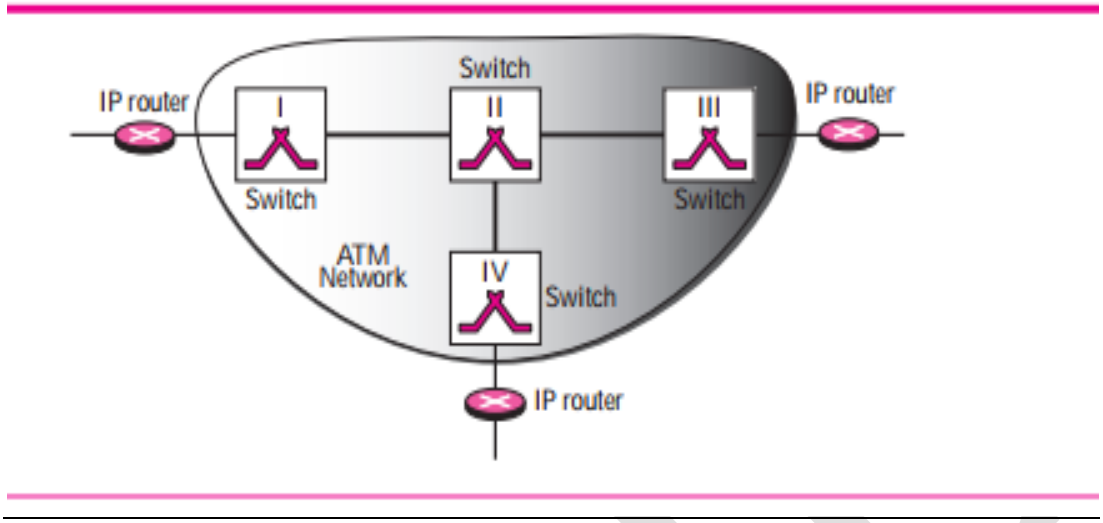


Physical Layer

The physical layer defines the transmission medium, bit transmission, encoding, and electrical to optical transformation. It provides convergence with physical transport protocols, such as SONET and T-3, as well as the mechanisms for transforming the flow of cells into a flow of bits.

ATM WANs

ATM, a cell-switched network, can be a highway for an IP datagram. Figure shows how an ATM network can be used in the Internet.



AAL Layer

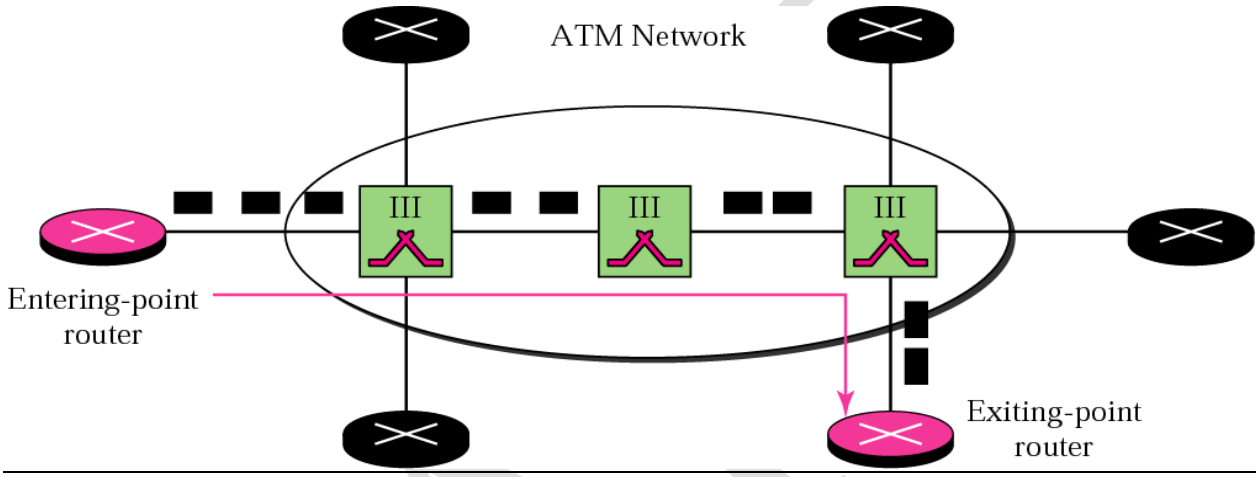
The only AAL used by the Internet is AAL5. It is sometimes called the simple and efficient adaptation layer (SEAL). AAL5 assumes that all cells created from one IP datagram belong to a single message. AAL5 therefore provides no addressing, sequencing, or other header information. Instead, only padding and a four-field trailer are added to the IP packet. AAL5 accepts an IP packet of no more than 65,536 bytes and adds an 8-byte trailer as well as any padding required to ensure that the position of the trailer falls where the receiving equipment expects it (at the last 8 bytes of the last cell). Once the padding and trailer are in place, AAL5 passes the message in 48-byte segments to the ATM layer.

Why Use AAL5?

A question that frequently comes up is why do we use AAL5. Why can't we just encapsulate an IP packet in a cell? The answer is that it is more efficient to use AAL5. If an IP datagram is to be encapsulated in a cell, the data at the IP level must be 53 − 5 − 20 = 27 bytes because a minimum of 20 bytes is needed for the IP header and 5 bytes is needed for the ATM header. The efficiency is 27/53, or almost 51 percent. By letting an IP datagram span over several cells, we are dividing the IP overhead (20 bytes) among those cells and increasing efficiency.

Routing the Cells

The ATM network creates a route between two routers. We call these routers enteringpoint and exiting-point routers. The cells start from the entering-point router and end at the exiting-point router as shown in Figure.



Addresses

Routing the cells from one specific entering-point router to one specific exiting-point router requires three types of addressing: IP addresses, physical addresses, and virtual circuit identifiers.

IP Addresses

Each router connected to the ATM network has an IP address. Later we will see that the addresses may or may not have the same prefix. The IP address defines the router at the IP layer. It does not have anything to do with the ATM network.

Physical Addresses

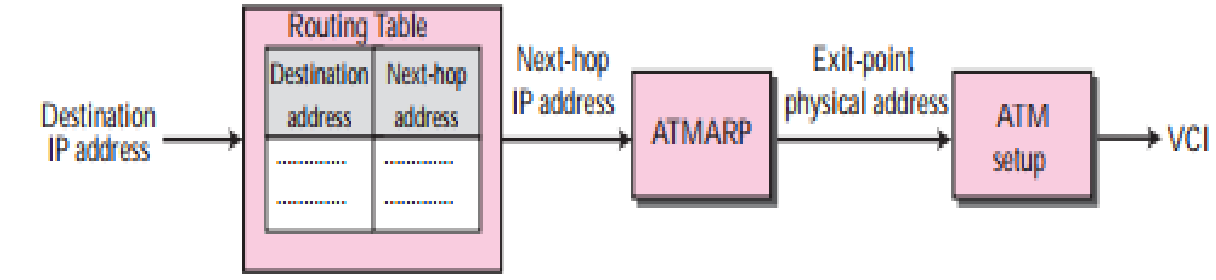
Each router (or any other device) connected to the ATM network has also a physical address. The physical address is associated with the ATM network and does not have anything to do with the Internet. The ATM Forum defines 20-byte addresses for ATM networks. Each address must be unique in a network and is defined by the network administrator. The physical addresses in an ATM network play the same role as the MAC addresses in a LAN. The physical addresses are used during connection establishment.

Virtual Circuit Identifiers

The switches inside the ATM network route the cells based on the virtual circuit identifiers (VPIs and VCIs), as we discussed in Chapter 3. The virtual circuit identifiers are used during data transfer.

Address Binding

An ATM network needs virtual circuit identifiers to route the cells. The IP datagram contains only source and destination IP addresses. Virtual circuit identifiers must be determined from the destination IP address. Figure shows how this is done.



These are the steps:

1. The entering-point router receives an IP datagram. It uses the destination address and its routing table to find the IP address of the next router, the exiting-point router. This is exactly the same step followed when a datagram passes through a LAN.

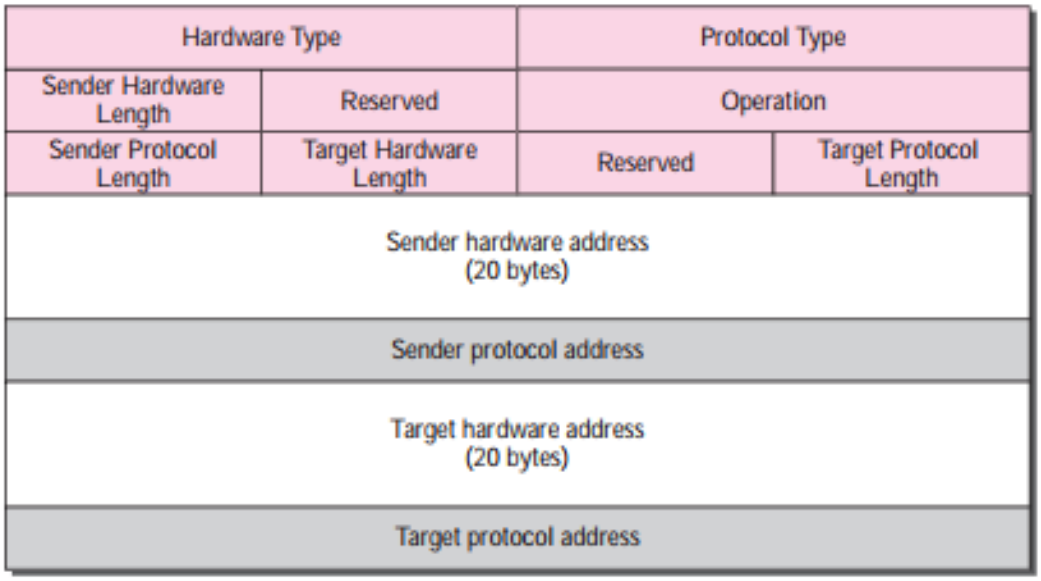
2. The entering-point router uses the services of a protocol called ATMARP to find the physical address of the exiting-point router. ATMARP is similar to ARP .

3. The virtual circuit identifiers are bound to the physical addresses

ATMARP

When IP packet are moving through an ATM WAN, a mechanism protocol is needed to find (map) the physical address of the exiting-point router in the ATM WAN given the IP address of the router. This is the same task performed by ARP on a LAN. However, there is a difference between a LAN and an ATM network.

A LAN is a broadcast network (at the data link layer); ARP uses the broadcasting capability of a LAN to send (broadcast) an ARP request. An ATM network is not a broadcast network; another solution is needed to handle the task. Packet Format The format of an ATMARP packet, which is similar to the ARP packet, is shown in Figure .



.

The fields are as follows:

Hardware type (HTYPE). The 16-bit HTYPE field defines the type of the physical network. Its value is 001316 for an ATM network.

Protocol type (PTYPE). The 16-bit PTYPE field defines the type of the protocol. For IPv4 protocol the value is 080016.

Sender hardware length (SHLEN). The 8-bit SHLEN field defines the length of the sender's physical address in bytes. For an ATM network the value is 20. Note that if the binding is done across an ATM network and two levels of hardware addressing are necessary, the neighboring 8-bit reserved field is used to define the length of the second address.

Operation (OPER). The 16-bit OPER field defines the type of the packet. Five packet types are defined as shown in Table

Message	*OPER value*
Request	1
Reply	2
Inverse Request	8
Inverse Reply	9
NACK	10

Sender protocol length (SPLEN). The 8-bit SPLEN field defines the length of the address in bytes. For IPv4 the value is 4 bytes.

Target hardware length (TLEN). The 8-bit TLEN field defines the length of the receiver's physical address in bytes. For an ATM network the value is 20. Note that if the binding is done across an ATM network and two levels of hardware addressing are necessary, the neighboring 8-bit reserved field is used to define the length of the second address.

Target protocol length (TPLEN). The 8-bit TPLEN field defines the length of the address in bytes. For IPv4 the value is 4 bytes.

Sender hardware address (SHA). The variable-length SHA field defines the physical address of the sender. For ATM networks defined by the ATM Forum, the length is 20 bytes.

Sender protocol address (SPA). The variable-length SPA field defines the address of the sender. For IPv4 the length is 4 bytes.

Target hardware address (THA). The variable-length THA field defines the physical address of the receiver. For ATM networks defined by the ATM Forum, the length is 20 bytes. This field is left empty for request messages and filled in for reply and NACK messages.

Target protocol address (TPA). The variable-length TPA field defines the address of the receiver. For IPv4 the length is 4 bytes.

ATMARP Operation

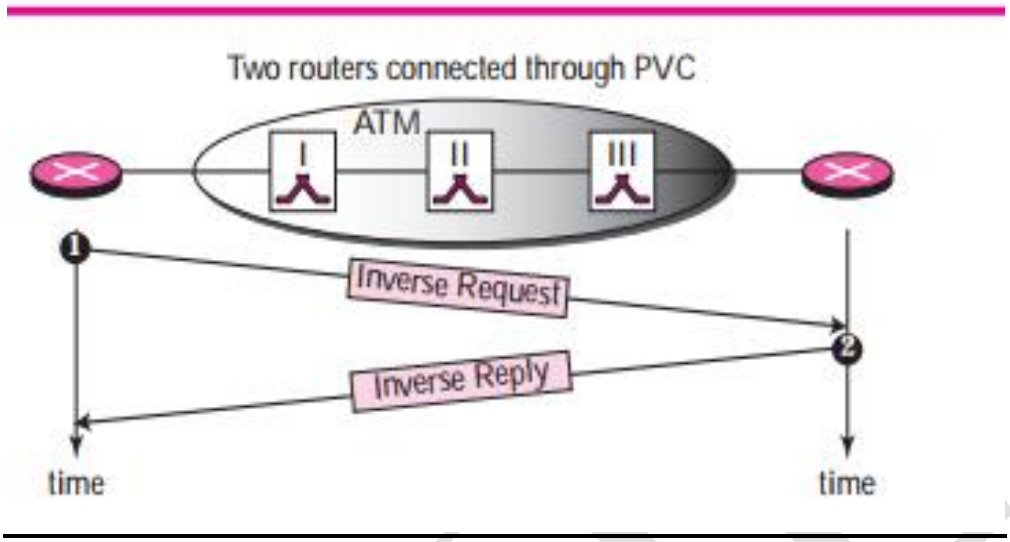
There are two methods to connect two routers on an ATM network: through a permanent virtual circuit (PVC) or through a switched virtual circuit (SVC). The operation of ATMARP depends on the connection method.

PVC Connection

A permanent virtual circuit (PVC) connection is established between two end points by the network provider. The VPIs and VCIs are defined for the permanent connections and the values are entered in a table for each switch.

If a permanent virtual circuit is established between two routers, there is no need for an ATMARP server. However, the routers must be able to bind a physical address to an IP address. The inverse request message and inverse reply message can be used for the binding. When a PVC is established for a router, the router sends an inverse request message.

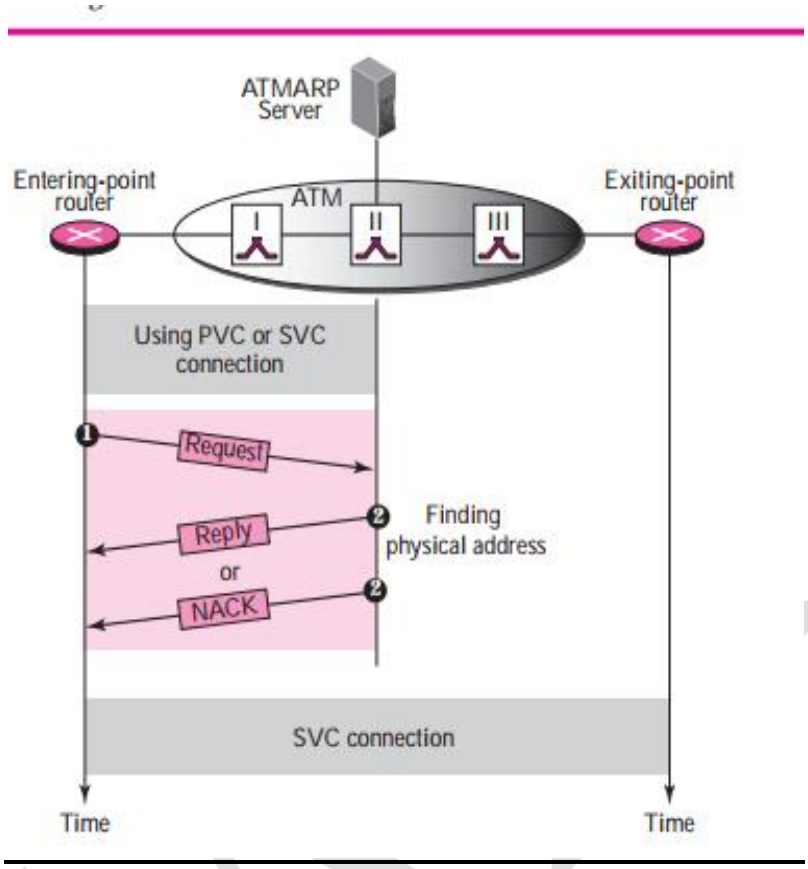
The router at the other end of the connection receives the message (which contains the physical and IP address of the sender) and sends back an inverse reply message (which contains its own physical and IP address). After the exchange, both routers add a table entry that maps the physical addresses to the PVC. Now, when a router receives an IP datagram, the table provides information so that the router can encapsulate the datagram using the virtual circuit identifier. Figure shows the exchange of messages between two routers.



SVC Connection

In a switched virtual circuit (SVC) connection, each time a router wants to make a connection with another router (or any computer), a new virtual circuit must be established. However, the virtual circuit can be created only if the entering-point router knows the physical address of the exiting-point router (ATM does not recognize IP addresses).

To map the IP addresses to physical addresses, each router runs a client ATMARP program, but only one computer runs an ATMARP server program. To understand the difference between ARP and ATMARP, remember that ARP operates on a LAN, which is a broadcast network. An ARP client can broadcast an ARP request message and each router on the network will receive it; only the target router will respond. ATM is a nonbroadcast network; an ATMARP request cannot reach all routers connected to the network. The process of establishing a virtual connection requires three steps: connecting to the server, receiving the physical address, and establishing the connection. Figure shows the steps.



Connecting to the Server

Normally, there is a permanent virtual circuit established between each router and the server. If there is no PVC connection between the router and the server, the server must at least know the physical address of the router to create an SVC connection just for exchanging ATMARP request and reply messages.

Receiving the Physical Address

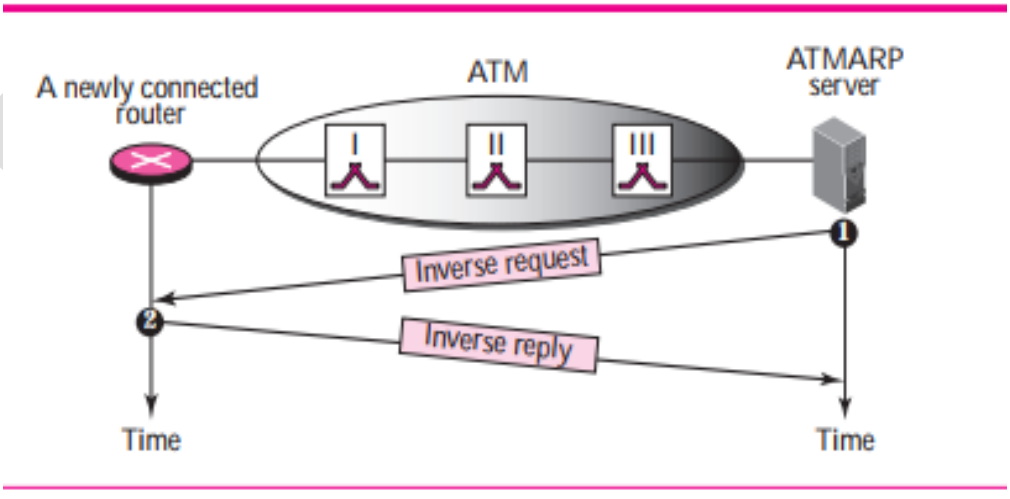
When there is a connection between the enteringpoint router and the server, the router sends an ATMARP request to the server. The server sends back an ATMARP reply if the physical address can be found or an ATMARP NACK otherwise. If the entering-point router receives a NACK, the datagram is dropped.

Establishing Virtual Circuits

After the entering-point router receives the physical address of the exiting-point router, it can request an SVC between itself and the exitingpoint router. The ATM network uses the two physical addresses to set up a virtual circuit which lasts until the entering-point router asks for disconnection. In this step, each switch inside the network adds an entry to its tables to enable them to route the cells carrying the IP datagram.

Building the Table

This is done through the use of ATMARP and the two inverse messages (inverse request and inverse reply). When a router is connected to an ATM network for the first time and a permanent virtual connection is established between the router and the server, the server sends an inverse request message to the router. The router sends back an inverse reply message, which includes its IP address and physical address. Using these two addresses, the server creates an entry in its routing table to be used if the router becomes an exiting-point router in the future. Figure shows the inverse operation of ATMARP.

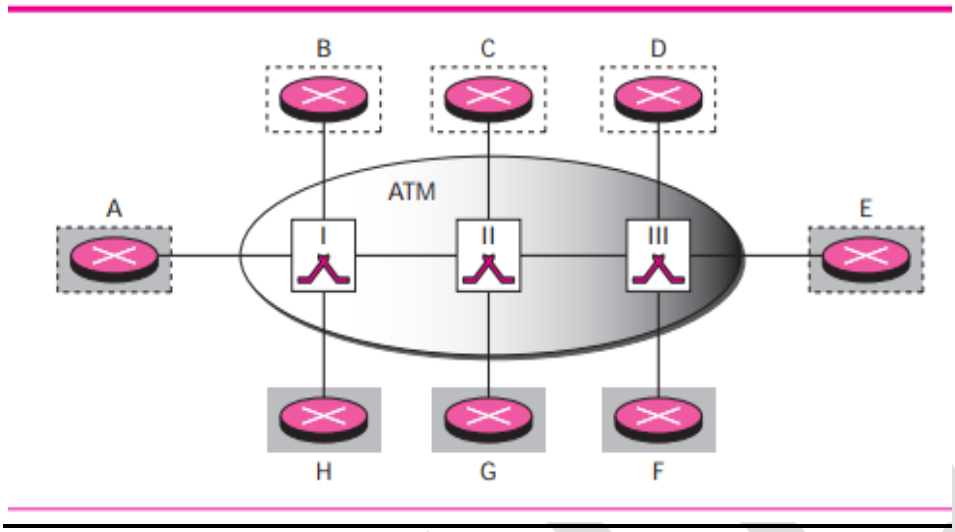


Logical IP Subnet (LIS)

Before we leave the subject of IP over ATM, we need to discuss a concept called logical IP subnet (LIS). For the same reason that a large LAN can be divided into several subnets, an ATM network can be divided into logical (not physical) subnetworks. This facilitates the operation of ATMARP and other protocols (such as IGMP) that need to simulate broadcasting on an ATM network. Routers connected to an ATM network can belong to one or more logical subnets, as shown in Figure.

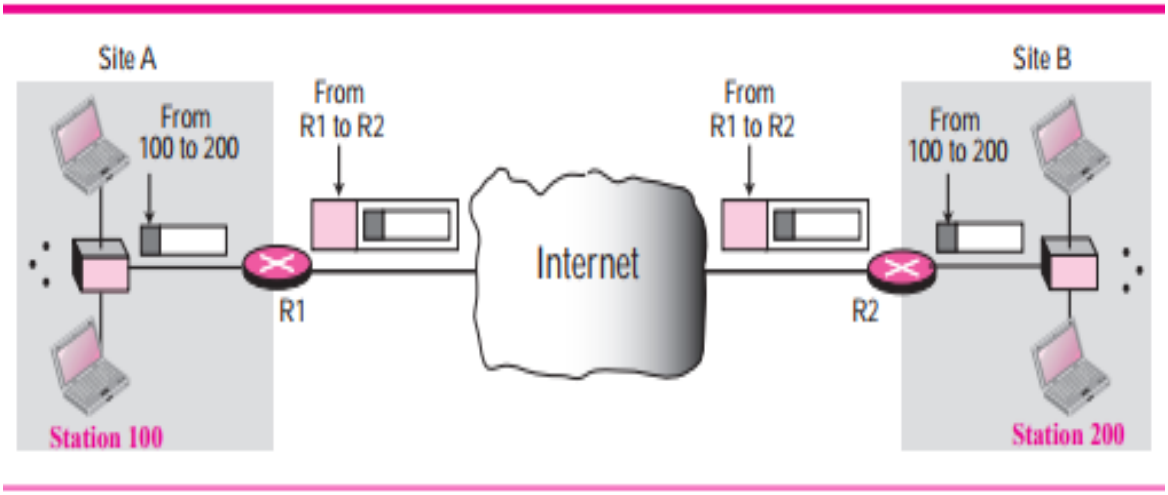
In the figure, routers B, C, and D belong to one logical subnet (shown by broken-line boxes); routers F, G, and H belong to another logical subnet (shown by shaded boxes). Routers A and E belong to both logical subnets.

A router can communicate and send IP packets directly to a router in the same subnet; however, if it needs to send a packet to a router that belongs to another subnet, the packet must first go to a router that belongs to both subnets. For example, router B can send a packet directly to routers C and D. But a packet from B to F must first pass through A or E. Note that routers belonging to the same logical subnet share the same prefix and subnet mask. The prefix for routers in different subnets is different. To use ATMARP, there must be a different ATMARP server in each subnet. For example, in the above figure, we need two ATMARP servers, one for each subnet.



VPN-Virtual Private Network

One of the applications of IPsec is in virtual private networks. A virtual private network (VPN) is a technology that is gaining popularity among large organizations that use the global Internet for both intra- and inter-organization communication, but require privacy in their intra-organization communication. VPN is a network that is private but virtual. It is private because it guarantees privacy inside the organization. It is virtual because it does not use real private WANs; the network is physically public but virtually private. Figure shows the idea of a virtual private network. Routers R1 and R2 use VPN technology to guarantee privacy for the organization.



VPN technology uses ESP protocol of IPSec in the tunnel mode. A private datagram, including the header, is encapsulated in an ESP packet. The router at the border of the sending site uses its own IP address and the address of the router at the destination site in the new datagram. The public network (Internet) is responsible for carrying the packet from R1 to R2. Outsiders cannot decipher the contents of the packet or the source and destination addresses. Deciphering takes place at R2, which finds the destination address of the packet and delivers it.

VPN Technologies

A well-designed VPN uses several methods in order to keep your connection and data secure.

- **Data Confidentiality**—This is perhaps the most important service provided by any VPN implementation. Since your private data travels over a public network, data confidentiality is vital and can be attained by encrypting the data. This is the process of taking all the data that one computer is sending to another and encoding it into a form that only the other computer will be able to decode.

  Most VPNs use one of these protocols to provide encryption.

  - **IPsec**—Internet Protocol Security Protocol (IPsec) provides enhanced security features such as stronger encryption algorithms and more comprehensive authentication. IPsec has two

encryption modes: tunnel and transport. Tunnel mode encrypts the header and the payload of each packet while transport mode only encrypts the payload. Only systems that are IPsec-compliant can take advantage of this protocol. Also, all devices must use a common key or certificate and must have very similar security policies set up.

For remote-access VPN users, some form of third-party software package provides the connection and encryption on the users PC. IPsec supports either 56-bit (single DES) or 168-bit (triple-DES) encryption.

- **PPTP/MPPE**—PPTP was created by the PPTP Forum, a consortium which includes US Robotics, Microsoft, 3COM, Ascend, and ECI Telematics. PPTP supports multi-protocol VPNs, with 40-bit and 128-bit encryption using a protocol called Microsoft Point-to-Point Encryption (MPPE). It is important to note that PPTP by itself does not provide data encryption.
- **L2TP/IPsec**—Commonly called L2TP over IPsec, this provides the security of the IPsec protocol over the tunneling of Layer 2 Tunneling Protocol (L2TP). L2TP is the product of a partnership between the members of the PPTP forum, Cisco, and the Internet Engineering Task Force (IETF). Primarily used for remote-access VPNs with Windows 2000 operating systems, since Windows 2000 provides a native IPsec and L2TP client. Internet Service Providers can also provide L2TP connections for dial-in users, and then encrypt that traffic with IPsec between their access-point and the remote office network server.

- **Data Integrity**—While it is important that your data is encrypted over a public network, it is just as important to verify that it has not been changed while in transit. For example, IPsec has a mechanism to ensure that the encrypted portion of the packet, or the entire header and data portion of the packet, has not been tampered with. If tampering is detected, the packet is dropped. Data integrity can also involve authenticating the remote peer.
- **Data Origin Authentication**—It is extremely important to verify the identity of the source of the data that is sent. This is necessary to guard against a number of attacks that depend on spoofing the identity of the sender.

- **Anti Replay**—This is the ability to detect and reject replayed packets and helps prevent spoofing.
- **Data Tunneling/Traffic Flow Confidentiality**—Tunneling is the process of encapsulating an entire packet within another packet and sending it over a network. Data tunneling is helpful in cases where it is desirable to hide the identity of the device originating the traffic. For example, a single device that uses IPsec encapsulates traffic that belongs to a number of hosts behind it and adds its own header on top of the existing packets. By encrypting the original packet and header (and routing the packet based on the additional layer 3 header added on top), the tunneling device effectively hides the actual source of the packet. Only the trusted peer is able to determine the true source, after it strips away the additional header and decrypts the original header. As noted in RFC 2401 , "...disclosure of the external characteristics of communication also can be a concern in some circumstances. Traffic flow confidentiality is the service that addresses this latter concern by concealing source and destination addresses, message length, or frequency of communication. In the IPsec context, using ESP in tunnel mode, especially at a security gateway, can provide some level of traffic flow confidentiality." All the encryption protocols listed here also use tunneling as a means to transfer the encrypted data across the public network. It is important to realize that tunneling, by itself, does not provide data security. The original packet is merely encapsulated inside another protocol and might still be visible with a packet-capture device if not encrypted. It is mentioned here, however, since it is an integral part of how VPNs function.

Tunneling requires three different protocols.

- **Passenger protocol**—The original data (IPX, NetBeui, IP) that is carried.
- **Encapsulating protocol**—The protocol (GRE, IPsec, L2F, PPTP, L2TP) that is wrapped around the original data.
- **Carrier protocol**—The protocol used by the network over which the information is traveling.

The original packet (Passenger protocol) is encapsulated inside the encapsulating protocol, which is then put inside the carrier protocol's header (usually IP) for transmission over the public network. Note that the encapsulating protocol also quite often carries out the encryption of the data. Protocols such as IPX and NetBeui, which would normally not be transferred across the Internet, can safely and securely be transmitted.

For site-to-site VPNs, the encapsulating protocol is usually IPsec or Generic Routing Encapsulation (GRE). GRE includes information on what type of packet you are encapsulating and information about the connection between the client and server.

For remote-access VPNs, tunneling normally takes place using Point-to-Point Protocol (PPP). Part of the TCP/IP stack, PPP is the carrier for other IP protocols when communicating over the network between the host computer and a remote system. PPP tunneling will use one of PPTP, L2TP or Cisco's Layer 2 Forwarding (L2F).

- **AAA**—Authentication, authorization, and accounting is used for more secure access in a remote-access VPN environment. Without user authentication, anyone who sits at a laptop/PC with pre-configured VPN client software can establish a secure connection into the remote network. With user authentication however, a valid username and password also has to be entered before the connection is completed. Usernames and passwords can be stored on the VPN termination device itself, or on an external AAA server, which can provide authentication to numerous other databases such as Windows NT, Novell, LDAP, and so on.
  When a request to establish a tunnel comes in from a dial-up client, the VPN device prompts for a username and password. This can then be authenticated locally or sent to the external AAA server, which checks:
  - Who you are (Authentication)
  - What you are allowed to do (Authorization)
  - What you actually do (Accounting)

The Accounting information is especially useful for tracking client use for security auditing, billing or reporting purposes.

- **Nonrepudiation**—In certain data transfers, especially those related to financial transactions, nonrepudiation is a highly desirable feature. This is helpful in preventing situations where one end denies having taken part in a transaction. Much like a bank requires your signature before honoring your check, nonrepudiation works by attaching a digital signature to the sent message, thus precluding the possibility of sender denying participation in the transaction.

POSSIBLE QUESTIONS

TWO MARKS

1. Define FTP.
2. What is the use of SMTP?
3. Define Anonymous FTP
4. Mention the commands and responses in SMTP.
5. What is SNMP?
6. Mention the purpose of MIB.
7. What is AAL layer?
8. How do you route the cells?
9. What is ATM?
10. What are the fields in ATMARP?
11. Mention the benefits of VPN.

EIGHT MARKS

1. Discuss about connections and communications in FTP.
2. Write a note on command processing in FTP.
3. Brief about commands and responses in SMTP.
4. Write about the operation of SNMP.
5. Explain the operation of ATM WAN.

6. Write the operation of ATM WAN.
7. Discuss about routing and carrying datagram in cells.
8. Write in brief about ATMARP.
9. Write short notes on logical IP subnet and ATMARP operation.
10. Explain techniques to guarantee privacy for an organization in VPN technology.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

ONE MARK QUESTIONS

DEPARTMENT OF COMPUTER SCIENCE

STAFF NAME: S.MANJU PRIYA **SUBJECT NAME: INTERNETWORKING WITH**

SUB.CODE: 17CSP201 **UNIT V** **SEMESTER: II**

S.NO	Question	Choice1	Choice2	Choice3	Choice4	Ans
1	___ allows organizations to use the global internet for private and public communication.	VPN	DNS	FTP	SNMP	VPN
2	A common technique to encrypt and authenticate in VPN is _____ .	internet security	IP security	network security	web security	IP security
3	____ involves the encapsulation of an encrypted IP datagram in a second outer datagram.	Ipsec	VPN	tunneling	none	tunneling
4	Mobile IP is an enhanced version of _____	TCP	IP	ARP	FTP	IP
5	The protocol that binds IP address and physical address is _____.	RARP	ATMWAN	ATMRP	ATMARP	ATMARP

6	_____ is a cell- switched network that can be a highway for an IP datagram.	ATM	SNMP	DNS	DHCP	ATM
7	_____ router receives an IP datagram.	exit-point	middle-point	entering-point	ending-point	entering-point
8	_____ connection is established between two end points by the network provider.	PVC	TPA	SHA	VPN	PVC
9	In a _____ connection each time a router wants to make a connection	PVC	SVC	SHA	none	SVC
10	An ATM network can be divided into _____ subnetworks.	physical	dynamic	static	logical	logical
11	In mobile IP the host has its orginal address called _____ address.	physical	logical	home	care-of	home
12	In mobile IP the host has its temporary address is called _____ address.	home	care of	physical	logical	care of
13	In mobile IP the care of address is associated with the ____ network.	home	LAN	foreign	All the above	foreign
14	_____ is usually a router attached to the home network of the mobile host.	home agent	foreign agent	mobile agent	none	home agent
15	_____ is usually a router attached to the foreign network of the mobile host.	home agent	foreign agent	mobile agent	none	foreign agent
16	When the mobile host act as a foreign agent called a _____ care of addresss.	located	allocated	co-located	none	co-located
17	The first phase in mobile communication is _____ discovery.	agent	user	server	client	agent

18	Mobile IP uses the router solicitation packet of _____.	IMCP	MICP	MCIP	ICMP	ICMP
19	_____ messages are encapsulated in a UDP user datagram.	resolution	registration	extension	identification	registration
20	_____ network private internet and access to the global internet.	hybrid	VPN	private	public	hybrid
21	The actual mail transfer is done through _____.	SDA	TTA	MTA	both a & b	MTA
22	The protocol that defines the MTA client and server in the internet is called __	SNMP	MTP	FTP	SMTP	SMTP
23	SMTP defines _____ commands.	13	14	24	15	14
24	_____ is a TCP/IP client- server application for copying files	FTP	SMTP	SNMP	ICMP	FTP
25	FTP uses _____ well- known TCP ports.	3	2	4	5	2
26	In FTP port _____ is used for the control connection.	20	21	22	23	21
27	In FTP port _____ is used for data connection.	20	21	22	23	20
28	In FTP the client sents port number to the server using _____ command.	PASV	PORT	LIST	OPEN	PORT
29	In FTP the control connection is made between _____ processes.	control	data transfer	data connection	data store	control

30	In FTP the data connection is made between _____ processes.	control	data transfer	data connection	none	data transfer
31	_____ is a general purpose client-server program.	NSFNET	CSNET	TELNET	ARPANET	CSNET
32	When a user logs into a local time sharing system it is called _____.	remote login	local login	security login	none	local login
33	When a user wants to access an application program he performs _____.	remote login	local login	security login	global login	remote login
34	___ driver pretends the characters are coming from a terminal.	non terminal	pseudo termainal	remote terminal	local termainal	pseudo termainal
35	Control characters can be used to handle _____ server.	local	global	remote	none	remote
36	In the _____ mode the client sends one character at a time to the server.	character	line	point	both a & c	character
37	In the _____ mode the client sends one line at a time to the server.	character	line	point	both a & c	line
38	A ___is a group of connected, communicating devices such asComputers and Printers.	Internet	Bridge	Network	Network.	Bridge
39	_____ Software provide communication between hosts	NCP	IMP	ACM	None	NCP
40	_____ protocol provide domain name services	Echo	daytime	name server	quote	name server
41	_____ protocol returns string of characters.	daytime	quote	chargen	RPC	daytime

42	The connection establishment in ____is called three way handshaking.	UDP	FTP	TCP	TCP/IP	FTP
43	ISN means _____.	Initial Sequence Number	Initial Service Number	Initial Segment Number	Initial Segment Node	Initial Sequence Number
44	_____ is a string of characters that hold some information	country domain	compression	cookie	none	cookie
45	FTP uses the service of _____	IP	TCP	SMTP	IGMP	TCP
46	In FTP Port 21 is used for _____	Control connection	Data connection	Transformation connection	one	Control connection
47	FTP uses_____ character set	NVTA ASCII	VTM ASCII	Binary	None	VTM ASCII
48	_____ mode data is delivered from FTP to TCP as continuous stream of bytes.	Block mode	Compress mode	Stream mode	None.	Compress mode
49	_____ Command terminates the message in SNMP	END	QUIT	LOOP	None	QUIT
50	_____ is a permanent negative completion reply	4YZ	5YZ	3YZ	YZ	3YZ
51	ATM has _____ formats for header	2	1	4	3	1
52	The switches inside he ATM Network route he cells on the _____	VPI	VCI	ARP	None	VPI
53	OPER is a _____bit field	8	45	12	34	45

54	SPA stands for _____	Sender Protocol Address	Seder Protocol Access	Sender Private Application	None	Sender Protocol Address
55	ATM accepts _____ bytes and transfers it into _____ bytes.	40 to 50	30 to 50	48 to 65	48 to 53.	48 to 65
56	First Phase in the Mobile Communication is _____	Agent advertisement	Agent finding	Agent discovery	None	Agent advertisement
57	_____ Network is designed to be used only inside an organization	Protected	Private	Sensitive	Non Sensitive	Sensitive
58	Private networks can provide ___for organizations	Efficiency	Privacy	A and B	None of the above	Privacy
59	Both private and hybrid networks have a major drawback: _____	Lack of privacy	Cost	Lack of security	Time consuming	Cost
60	VPN is a network that is _____ but _____.	private; public	private; virtual	Public ; virtual	None of the above	private; virtual
61	VPN is physically _____ but virtually _____.	public; private	private; virtual	Public ; virtual	None of the above	public; private
62	A VPN can use _____ to guarantee privacy	IP Sec	Tunneling	Both a and b	None	Tunneling
63	On a network that uses NAT, the _____ has a translation table.	Bridge	Router	Server	None of above	Router
64	On a network that uses NAT, _____ initiates the communication	An internal host	An External host	Router	Hub	An internal host
65	On a network that uses NAT, the router can use _____ global address	1	2	3	None of the above	1

TCP/IP