Module I

1

NETWORKING - I

Unit Structure

1.1 OBJECTIVE

1. To help to get a grounding of basic network components and architecture.
2. To explore basic networking models.

3. To learn the way protocols are used in networks.
4. Understanding of the fundamental concepts of computer networking.
5. To understand the basic taxonomy and terminology of the computer networking area.
6. To understand the Internet and Intranet, Protocol layer and their services, Network Applications like web, HTTP, FTP and Electronic Mail in the Internet, Domain Name System, and Transport-Layer Services.

1.2 INTRODUCTION

Computer Network is potentially the largest skillfully arranged system ever created by people, with hundreds of millions of connected systems, communication links, and switches with billions of users who connect via laptops, tablets, and smart phones and with an array of new Internet-connected devices such as sensors, webcams, game consoles, picture frames, and even washing machines etc. The systems are connected to each other's by a network communication links and packet switches; those are many types of communication links, which are made up of different types of physical media, like coaxial cable, copper wire, optical fiber, and radio spectrum. Various links can transmit data at different rates, with the transmission rate of a link measured in bits per second. When one system has data to send to another system, the sending system segments the data and adds header of bytes to each segment.

1.3 INTERNET AND INTRANET

The Internet is a worldwide system of interconnected computer networks. It uses the standard Internet Protocol (TCP/IP), set of rules. Each and Every computer on the Internet is identified by a unique IP address. IP Address is a unique set of numbers (such as 110.22.33.114) which identifies a computer's physical location. The Internet is accessible to every user all over the world.

A special computer Domain Name Server is used to provide a name to the IP Address so that the user can locate & identify a computer by a name. For example, a DNS server will resolve a name https://www.mu.ac.in to a particular IP address to uniquely identify the computer on which this website is hosted.

Intranet is the system in which multiple computers are connected to each other's through physical media. Systems on the intranet are not available to the world outside the intranet. Normally each organization has its own Intranet network and members of that organization can access the computers in their intranet. Each computer in Intranet is also identified by an IP Address which is unique in between the computers in that Intranet.

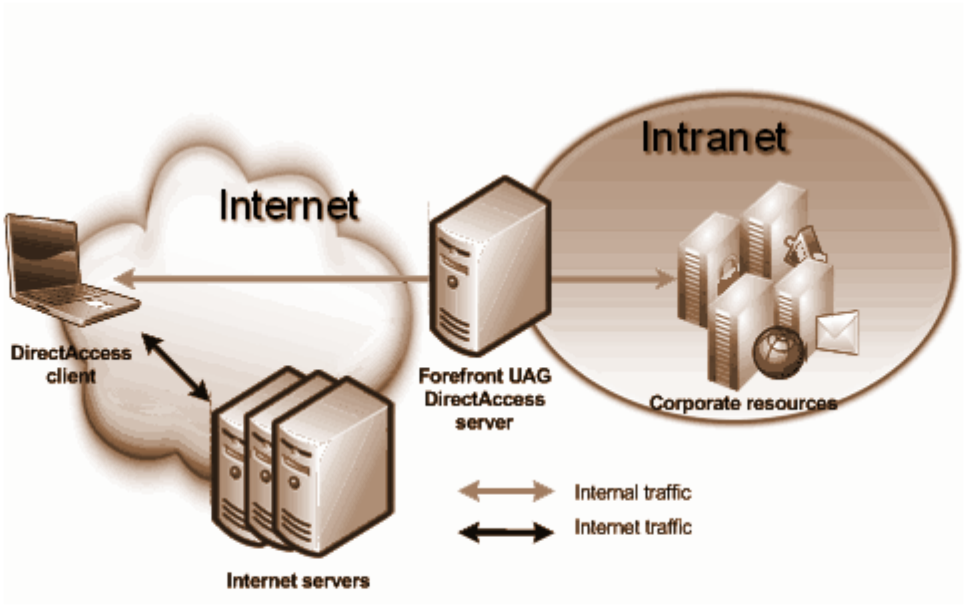


Fig.1 Internet & Intranet

Similarities between Internet and Intranet

- Intranet uses the internet protocols (IP) such as TCP/IP and FTP.
- Intranet sites are accessible via the web browser in a similar way as websites on the internet. However, only members of Intranet network can permit access to intranet hosted sites.
- In Intranet, own instant messengers can be used over the internet.

Differences between Internet and Intranet

- The Internet is general to Computers all over the world whereas Intranet is distinct to few computers.
- The Internet provides a wider and better access to websites to a large population, whereas Intranet is restricted.
- The Internet is not as safe as the Internet. Intranet can be safely privatized as per the need.

1.4 PROTOCOL LAYER AND THEIR SERVICES

A protocol is required when two entities need to communicate with each other's. When communication is not easy or simple, it may divide the complex task of communication into different layers. On this occasion, we may need various protocols, one for each layer. Let us use a scenario in communication in which the role of protocol layering may be better accepted.

A **layered architecture** allows us to talk over a well explained, different part of a large and complex system. This simplification itself is of extraordinary value by providing modularity, making it much easier to change the implementation of the service provided by the layer. As long as the layer provides the same service to the layer above it, and uses the identical services from the layer below it, the balance of the system perseveres unchanged when a layer's implementation is changed.

A protocol layer such as HTTP and SMTP are almost implemented in software in the end systems, so are transport-layer protocols. The physical layer and data link layers are responsible for administering communication over a distinct link; they are typically implemented in a network interface card associated with a given link. The network layer is generally an associated implementation of hardware and software parts.

Application Layer

The application layer is a network application and their application-layer protocols reside on this layer. The Internet's application layer includes different protocols like, the HTTP protocol which provides for web document request and transfer, SMTP which administers for the transfer of e-mail messages, and FTP which provides for the transfer of files between two systems.

Transport Layer

The Internet's transport layer carries application-layer messages between application endpoints. In

The two transport protocols, TCP and UDP, any one of them can transport application layer messages to each other's. TCP provides a connection-oriented service to its applications. This includes guaranteed delivery of message or packets to the destination and flow control.

Network Layer

Network layer is responsible for moving network-layer packets known as datagrams from one host to another host. The Internet transport-layer protocol such as TCP & UDP in a source host passes a transport-layer segment and a destination address to the network layer, like the postal service a letter with a destination address. The network layer provides the service of delivering the chunk of message to the transport layer in the destination host.

Link Layer

The network layer directs a datagram through a list of routers between the source and destination. To move a packet from one router to the next route in the route, the network layer confides on the services of the link layer. At every node the network layer sends the datagram packet down to the link layer, which delivers the datagram packet to the next node along the route information. At this point next node, the link layer passes the datagram up to the network layer.

Physical Layer

In Physical Layer, a period of time the function of the link layer is to move entire frames from one network element to an adjacent network element; the job of the physical layer is to shift the individual bits within the frame from one node to the next node. The protocol link is dependent and furthers relies on the actual transmission medium of the protocol link, for example, twisted-pair copper wire, single-mode fiber optics. For e.g.

Ethernet has numerous physical-layer protocols, one for twisted-pair copper wire, another for coaxial cable, another for fiber, and so on.

1.5 NETWORK APPLICATIONS LIKE WEB, HTTP, FTP

Web

The web was the first Internet application that was used everywhere in today's world. It drastically & continuously changes how people collaborate inside and outside their work environment. It raised the Internet from just one of many data networks to essentially the one and only one data network. Maybe what appeals the most to users is that the web operates on requirement. Web users receive what they need, when they want it at any time. This is far from traditional broadcast radio and television systems, which force users to tune in when the content provider makes the content available to all users in the world. Web available on user demand. It is extremely simple for any individual to make information available over the web for every user can be a publishes at too little cost. Hyperlinks and search engines help us guide through information. Forms, JavaScript, Java applets, and many others devices empower us to relate with different web pages and websites. The web and its protocols assist as a platform for YouTube, web-based email, and most mobile Internet applications, including Instagram and Google Maps.

1.5.1 Overview of HTTP

.

The Hypertext Transfer Protocol (HTTP) is the web’s application-layer protocol, transmitting hypermedia documents and is at the heart of the web. HTTP is implemented in two programs:

a. a client program and
b. A server program.

The client and server program, executing on different machines, talk to each other’s by exchanging HTTP messages. HTTP defines the structure of these messages and how the client and server exchange the messages & information.

A web page consists of objects. An object is a file like HTML file, a JPEG image, a Java applet, or a video clip that is addressable by a single URL. Many web pages consist of a base HTML file and several referenced objects. For example, the web page consists of a HTML text file and around five JPEG image files, and then the web page has a total six objects, the base HTML file and the five image files. The HTML file relates to the others objects in the page with the objects URLs. Every URL has two parts: the host name of the server and the object’s path name.

For e.g. the URL has www.SmallSchool.edu for a hostname and /Department/picture.gif for a path name. Because web browsers, such as Internet Explorer and Firefox implement the client side of HTTP, in the context of the web. The words used for browser and client conversely as a web server, which implement the server side of HTTP, web objects, each addressable by a URL. Well liked web servers include Apache and Microsoft Internet Information Server.

HTTP states how web clients request web pages from web servers and how servers transfer web pages to clients. When a user requests a particular web page, the browser sends HTTP request messages and requests objects which are available on the server side. The server receives the requests via HTTP and responds with HTTP response messages that include the objects. HTTP uses TCP/IP as its underlying transport protocol. The HTTP client first begins a TCP connection with the server. After the connection is formed, the browser and the server processes access TCP through their socket interfaces.

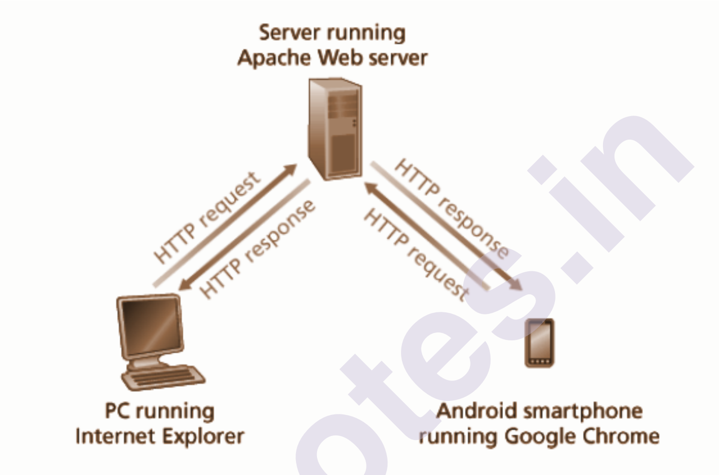


Fig.2 HTTP request & response

This inferred that each HTTP request message sent by a client process in time arrives intact at the server; similarly, each HTTP response message sent by the server process in time arrives intact at the client. The loss of data in HTTP need not doubt or TCP recovers from loss or reordering of data within the network system. That is the task of TCP and the protocols in the bottom layers of the protocol stack. It is compulsory to note that the server sends requested files to clients without storing any state information about the client. If a specific client asks for the same object twice in a time of a few seconds, the server does not respond by saying that it just provides the object to the client; instead, the server resends the object, as it has completely unknown what it did earlier. Because an HTTP server maintains no information about the clients, HTTP is said to be a stateless protocol in the network.

1.5.2 Persistent & Non-Persistent Connections

In most Internet applications, the client and server communicate for an extended period of time, with the client making a series of requests and the server responding to each of the requests on time one by one. based on & use of the application, the sequence of requests may be made one after the others back-to-back, from period to period, at regular intervals of time, or asymmetrical to gain a deep understanding of this design issue, let's examine the advantages and disadvantages of persistent connections in the context of a specific application, namely, both non-persistent connections and persistent connections use in HTTP. Even though in its default mode HTTP uses persistent connections, HTTP clients and servers can be well organized to use non-persistent connections instead.

1.5.3. HTTP with Non-Persistent Connections

Transferring of a web page from server machine to client machine in the case of no persistent connection, the web page contains a base HTML file and 10 JPEG images, and that all 11 of these objects are on the same server machine. For example the URL for the base HTML file is https://www.SmallSchool.edu/Department/home.index

a. The HTTP client process begins a TCP connection to the server www.SmallSchool.edu on port number 80, this is the default port number 80 used for HTTP. This TCP connection, those will be a socket both at the client and at the server.

b. Through socket, HTTP client sends an HTTP request message to the server. The request message includes the path name /Department/home. index which is shown above example.

c. The HTTP server processes the request message via its socket, retrieves the object /Department/home. index from its storage device i.e. from RAM or disk, enclose the object in an HTTP response message, and sends the response message to the client via socket.

d. The HTTP server process acknowledges TCP to close the TCP connection.

e. The HTTP client receives the response message from HTTP server. Then the TCP connection is terminated. The response message indicates that the encapsulated object is an HTML file. The client extracts this file from the response message, reads the HTML file, and finds sources to the 10 JPEG objects.

f. The first four steps i.e. a, b, c & d are repeated for each of the referenced JPEG objects. As the browser receives the web page at the client machine, it displays the page to the user on the client. Two different browsers may interpret a web page in somewhat different techniques. HTTP has neglected how a web page is interpreted by a client.

The steps above explain the use of non-persistent connections, whose each TCP connection is closed,

After the server sends the object, the connection does not persist for others objects. Each TCP connection transports exactly one request message and one response message. Thus, in this example, when a user requests the web page, 11 TCP connections are generated.

The Round Trip Time (RTT) includes packet-propagation delays, packet stream delays in between routers and switches, and packet-processing delays. In Figure 3, shows that to initiate a TCP connection between the client browser and the web server, this process requires a “three-way handshake” method. The client sends a small TCP segment to the server, the server acknowledges that segment and responds with a small TCP segment back, and finally the client sends acknowledgement back to the server. The first two sections of the three-way handshake take one RTT. After completing the first two parts of the handshake method, the client sends the HTTP request message combined with the third part of the three-way handshake into the TCP connection on the internet. Once the request message reaches the server, the server sends the HTML file into the TCP connection to the client browser. This HTTP request/response eats up another RTT. Thus, approximate, the total response time is two RTTs plus the transmission time at the server of the HTML file.

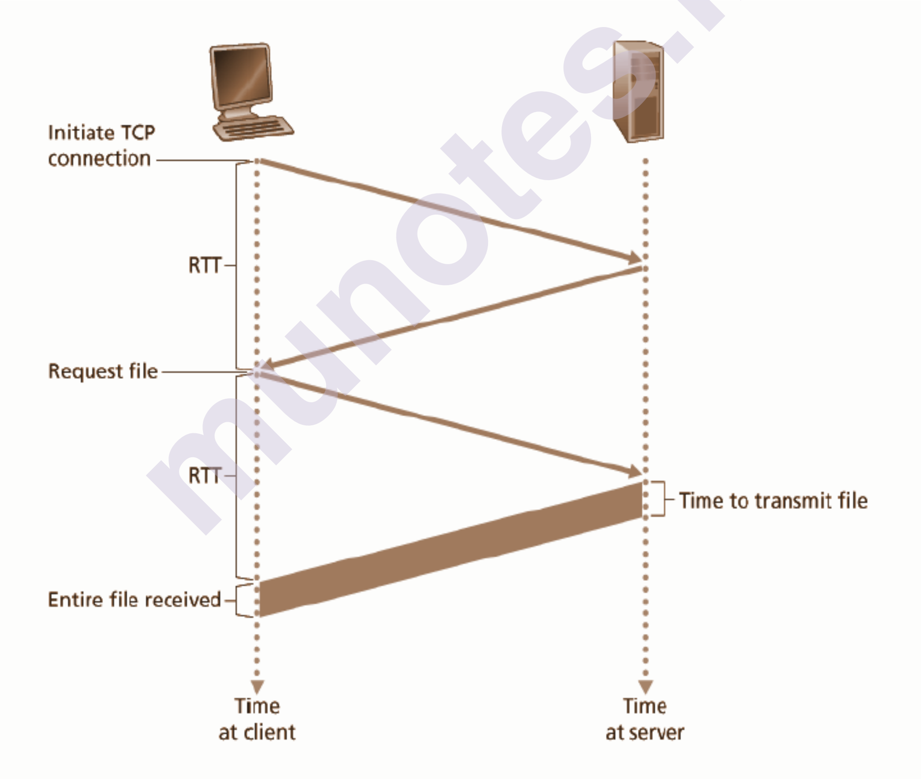


Fig. 3 Back-of-the-envelope calculation for the time needed to request and receive an HTML file

HTTP with Persistent Connections

After sending a response a server left TCP connection left open. Forthcoming requests and responses between the same client and server can be sent over the same connection. In some others cases, an entire web page can be sent over a single persistent TCP connection in HTTP. On the others hand, multiple web pages residing on the same server can be sent from the server machine to the same client machine over a single persistent TCP connection. These object requests can be made one after the others, without waiting for replies to pending requests. The HTTP server closes a connection when it isn't used for a few times. When the server receives the one after the others requests, it sends the objects back-to-back. The default mode of HTTP uses persistent connections with pipelining.

1.5.3 HTTP Message Format

The HTTP specifications include the definitions of the HTTP message formats. There are two types of HTTP messages, request messages and response messages.

HTTP request message:

```
GET /somedir/page.html HTTP/1.1
Host: www.SmallSchool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: french
```

This message is written in ASCII text, the human can easily read this. The second division of the message has five lines, each followed by a carriage return and a line feed. The end line of the message is followed by an additional carriage return and line feed. Even though this particular request message has five lines, a request message can have many more lines. The first line of an HTTP request message is called the request line and the future lines are called the header lines.

The three fields of request line:

1. Method field,
2. URL field,
3. HTTP version field.

The method field has various different values, including GET, POST, HEAD, PUT, and DELETE. The great most part of HTTP request messages have the GET method. The GET method is used when the browser requests an object from the server application, with the requested object is identified in the URL field.

The header line Host: www.SmallSchool.edu specifies the host on which the object resides. The host header line provides the information which is required by web proxy caches, including the connection: close header line, the browser instructs the server that it doesn't want to disturb persistent connections, and then the server closes the connection after sending the requested object to the client.

The User agent specifies the header line, that the browser type that is building the request to the server, at this moment the user agent is mozilla, a firefox browser. This header line is helpful because the server sends different varieties of the same object to different types of user agents.

The Accept-language header indicates that the user selects to receive a French version of the object, if such an object exists on the server; or else, the server should send its default version. The **Accept-language** header is just numerous content negotiation headers available in HTTP.

The general format of the request message, as shown in Figure 4. Beside the header lines those is an "entity body." The entity body is empty with the GET method, but is used with the POST method. HTTP clients generally use the POST method when the user fills out a form for e.g., when a user provides search words to a search engine. Using the POST message, the user is still requesting a web page from the server, but the specific contents of the web page depend on what the user entered into the form fields. The value of the POST method field is POST, and then the entity body contains what the user entered into the form fields.

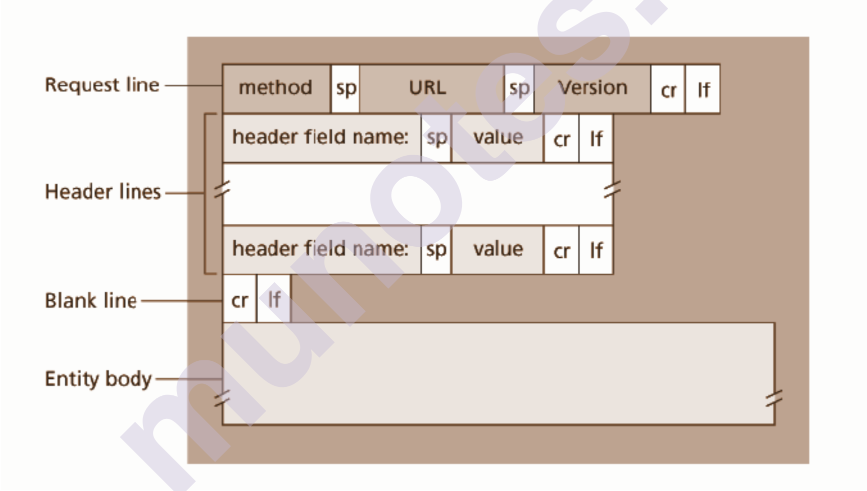


Fig. 4 General format of an HTTP request message

1.5.4 HTTP Response Message

HTTP response message. This response message could be the response to the example request message explained in HTTP Request Message.

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 30 Aug 2020 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 30 Aug 2020 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
(data data data data data ...)
```

The response Message has three sections: an initial status line, six header lines, and then the entity body. The entity body is the meat of the message; it contains the requested object itself.

The status line has three fields: the protocol version field, a status code, and a corresponding status message. In this example, the status line indicates that the server is using HTTP/1.1.

The header lines.

The server uses this connection for a close header line to inform the client that it is going to close the TCP connection after sending the message.

a. The Date header line shows the time and date when the HTTP response was created and sent by the server.

b. The Server header line shows that the message was generated by an apache web server.

c. User-agent the header line shows the HTTP request message.

d. Last- Modified the header line shows the time and date when the object was created or last modified.

e. Content-Length The header line shows the number of bytes of data in the object being sent.

f. Content-Type the header line shows that the object in the entity body is HTML text.

The general format of a response message:

The general format of HTTP response message which is shown in Figure 5 This general format of the response message is the same as the previous example of a response message format. The status code and associated phrase indicate the result of the request message. Some common status codes and associated phrases include:

- 200 OK status code shows that request succeeded and the information is returned in the response.
- 301 Moved Permanently status code requested object has been permanently moved & the new URL is specified in Location: header of the response message format. The client software will automatically fetch the new URL.
- 400 Bad Request is a generic error code status indicating that the request could not be known by the server.
- 404 Not found status code shows that the requested document does not exist on this server.
- 505 HTTP Version Not Supported by this message format: The requested HTTP protocol version is not supported by the server.

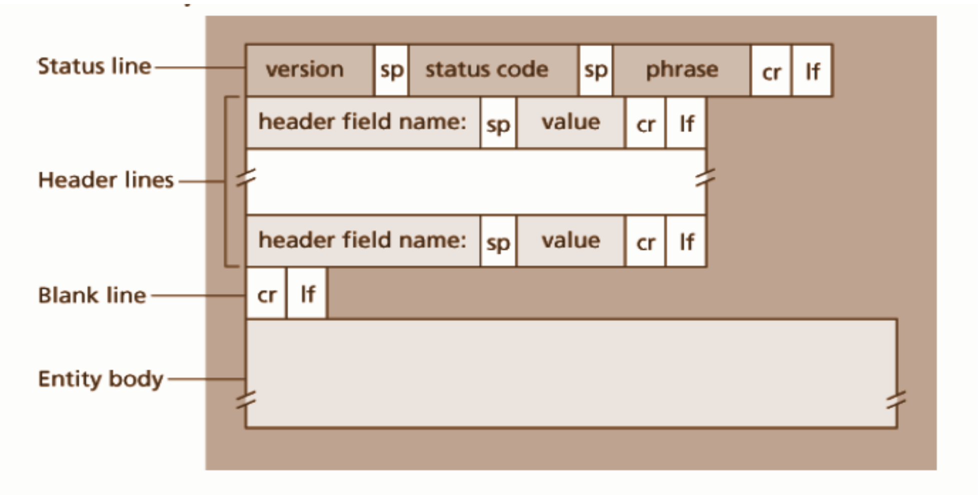


Fig. 5 General format of an HTTP response message

1.5.5 User-Server Interaction: Cookies

HTTP cookie is a small block of data stored on the user's computer or machine by the web browser while browsing a website. This clarifies server design and has permitted engineers to develop high-performance web servers that can handle thousands of concurrent TCP connections simultaneously. However, it is often advantageous for a web site to identify users, ethics because the server wishes to restrain user access or because it wants to accept content as a function of the user identity. For this reason cookies allow sites to keep track of all users. Most of the commercial web sites use cookies today.

Cookie technology has four components: (As shown in Figure 6)

(1) A cookie header line in the HTTP response message;
(2) A cookie header line in the HTTP request message;
(3) A cookie file kept on the user’s end system and managed by the user’s browser; and
(4) A back-end database at the web site.

As per the figure 6, an example of how cookies work. Assume amit, who always accesses the web using Internet Explorer from his home machine, visits Amazon.com sites for the first time. Let us suppose that in the past he has already visited the eBay website. When the request comes into the amazon web server, the server creates a unique identification number (UIN) and creates an entry in its backend database that is indexed by the identification number. The amazon web server then responds to amit’s web browser, also including in the HTTP response message a Set-cookie header, which consist of the identification number. For example, the header line is:

```
Set-cookie: 1678
```

When Amit’s browser receives the HTTP response message, it sees the Set-cookie is header. The browser then includes a line to the special cookie file that it manages. This line adds the host name of the server and the identification number in the Set-cookie header. Remember that the cookie file already has an entry for eBay, since Amit has visited that site in the past days. As Amit continues to browse the Amazon site, each time he requests a web page, his browser consults his cookie file, extracts his identification number for this site and places an identification number in the cookie header line in the HTTP request. Specifically, each of user HTTP requests to the Amazon server includes the header line:

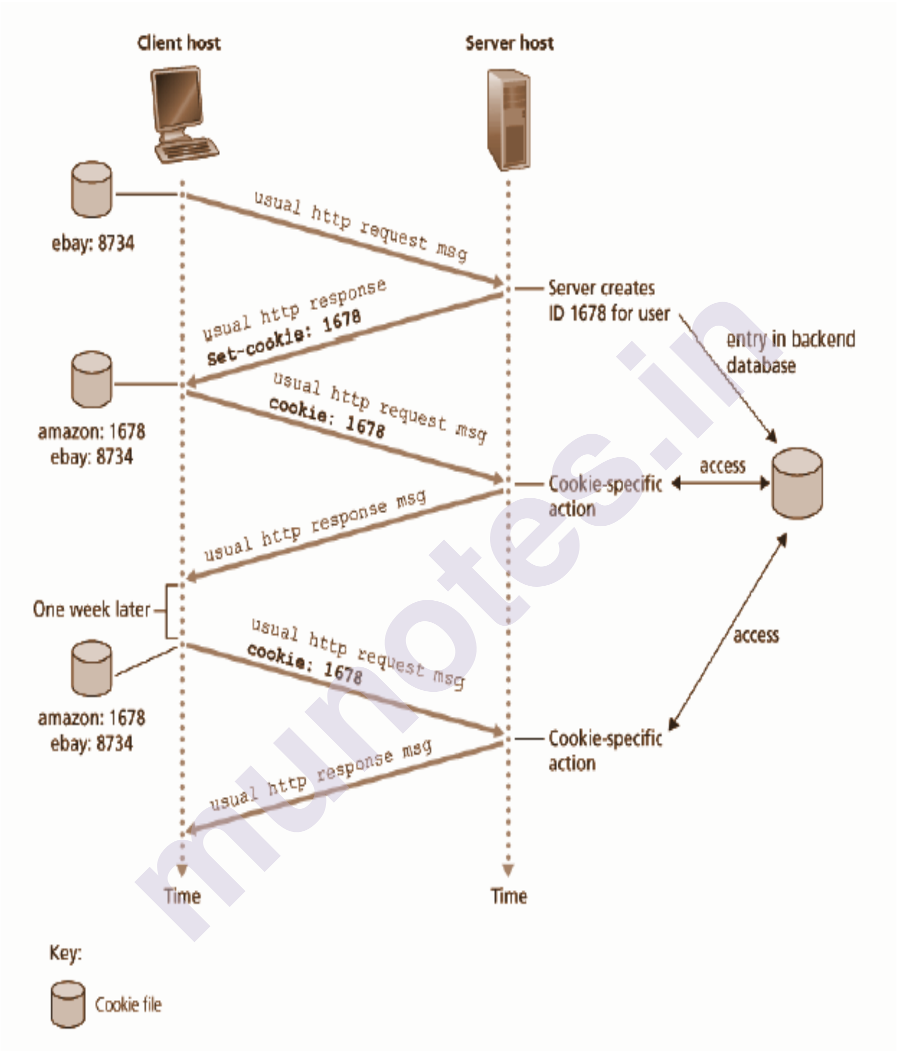


Fig. 6 keeping user state with cookies

```
Cookie: 1678
```

The amazon server is able to track Amit's activity at the Amazon site. Although the Amazon web site does not necessarily know Amit's name and its details, it knows exactly which pages user 1678 visited, in which order, and at what times. amazon uses cookies to provide its shopping cart service. Amazon can maintain a list of all of Amit's intended purchases, so that he can pay for them collectively at the end of the session. If amit returns to amazon's site, one week after, his browser will continue to put the header line Cookie: 1678 in the request messages. Amazon also recommends products to amit based on web pages he has visited at amazon in the past. If Amit also registers himself with amazon providing full name, e-mail address, postal address, and credit card information amazon can then include this information in its database, thereby associating Amit's name with his identification number. This is how Amazon and others e-commerce sites provide "one-click shopping" when Amit chooses to purchase an item during a subsequent visit, he doesn't need to re-enter his name, credit card number, or address.

1.6 FTP

In a typical FTP session the user is sitting in front of one host i.e. the local host and remote host. If the user to access the remote account, the user must provide a user identification and a password during file transfer. After providing authorization information, the user can transfer files from the local file system to the remote file system and vice versa. As shown in Figure 7, the user interacts with FTP through an FTP user agent. The user must provide the hostname of the remote host, causing the FTP client process in the local host to setup a TCP connection with the FTP server process in the remote host. The user then provides the user recognition and password, which are sent over the TCP connection as part of FTP. Once the server has authenticated the user, the user copies one or more files stored in the local file system into the remote file system or vice versa.

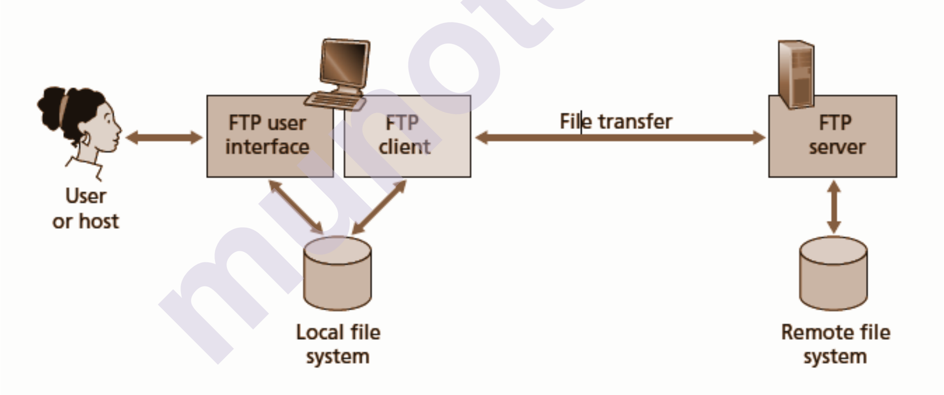


Fig. 7 FTP moves files between local and remote file systems

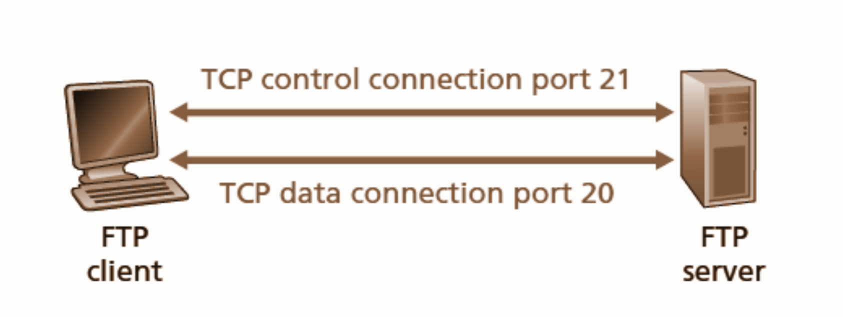


Fig. 8 Control and data connections

HTTP and FTP are both file transfer protocols and have many common characteristics:

1. They both run on top of TCP.

2. The most striking difference is that FTP uses two parallel TCP connections to transfer a file, a control connection and a data connection.

The FTP control and data connections are illustrated in Figure 8.

FTP Commands and Replies

The commands, from client to server, and replies, from server to client, are sent across the control connection in 7-bit ASCII format. Thus, like HTTP commands, FTP commands are readable by people. In order to delineate successive commands, a carriage return and line feed end each command. Each command consists of four uppercase ASCII characters, some with optional arguments. Some of the more common commands are given below:

- USER username: Used to send the user identification to the server.
- PASS password: Used to send the user password to the server.
- LIST: Used to ask the server to send back a list of all the files in the current remote directory. The list of files is sent over a (new and non-persistent) data connection rather than the control TCP connection.
- RETR filename: Used to retrieve a file from the current directory of the remote host. This command causes the remote host to initiate a data connection and to send the requested file over the data connection.
- STOR filename: Used to store a file into the current directory of the remote host.

Each command is followed by a reply, sent from server to client. The replies are three-digit numbers, with an optional message following the number. This is similar in structure to the status code and phrase in the status line of the HTTP response message. Some typical replies, along with their possible messages, are as follows:

- 331 Username OK, password required
- 125 Data connection already open; transfer starting
- 425 Can’t open data connection
- 452 Error writing file

1.7 ELECTRONIC MAIL IN THE INTERNET

Electronic mail has been around since the birth of the internet. It was the most popular application, and has become more improved and powerful over the years. It is the most important and utilized application. As it is the same like postal mail service, e-mail is an asynchronous communication medium people send and read messages when it is convenient for them, without having to interrelate with others people’s schedules. Electronic mail is fast, easy to distribute, and cheap. The current email application has many powerful features, including messages with attachments, hyperlinks, html-formatted text, embedded photos and much more. His application-layer protocols are the heart of internet e-mail.

A high-level view of the internet mail system and its key components:

Figure 1 shows a high-level view of the internet mail system. It has three major components: user agents, mail servers, and the simple mail transfer protocol (SMTP).

Example: amity, sending an e-mail message to a recipient, sham. User agents allow users to read, reply to, forward the message, save the message, and compose messages. Microsoft Outlook and Google Mail are examples of user agents for email applications. When it is finished composing a message, the user agent sends the message to amit mail server, whose message is placed in the mail server’s queue for outgoing messages. When shyam wants to read a message, his user agent recovers the message from his mailbox in his mail server. Each recipient, like shyam, has a mailbox located in one of the mail servers. Shyam's mailbox manages and maintains the messages that have been sent to him. A typical message starts its migration in the sender’s user agent, travels to the sender’s mail server, and travels to the recipient’s mail server, whose it is at stake in the recipient’s mailbox. When shyam wants to access the messages in his mailbox, the mail server containing his mailbox authenticates shyam. amit’s mail server must also deal with failures in shyam’s mail server. if amit’s server cannot deliver mail to shyam’s server, amit’s server holds the message in a message queue and attempts to transfer the message after some time. Reattempts for sending messages are often done every 30 minutes, if there is no success after several days, the server removes the message and notifies the sender i.e. to amit with an e-mail message SMTP is the principal application-layer protocol for internet electronic mail. It uses the reliable data transfer service of TCP to transfer mail from the sender’s mail server to the recipient’s mail server. In most application-layer protocols, SMTP has two sides, a client side, which executes on the sender’s mail server and a server side which executes on the recipient’s mail server. Both the client and server sides of SMTP run on every mail server. When a mail server sends mail to other mail servers, it acts as an SMTP client. When a mail server receives mail from others mail servers, it acts as an SMTP server.

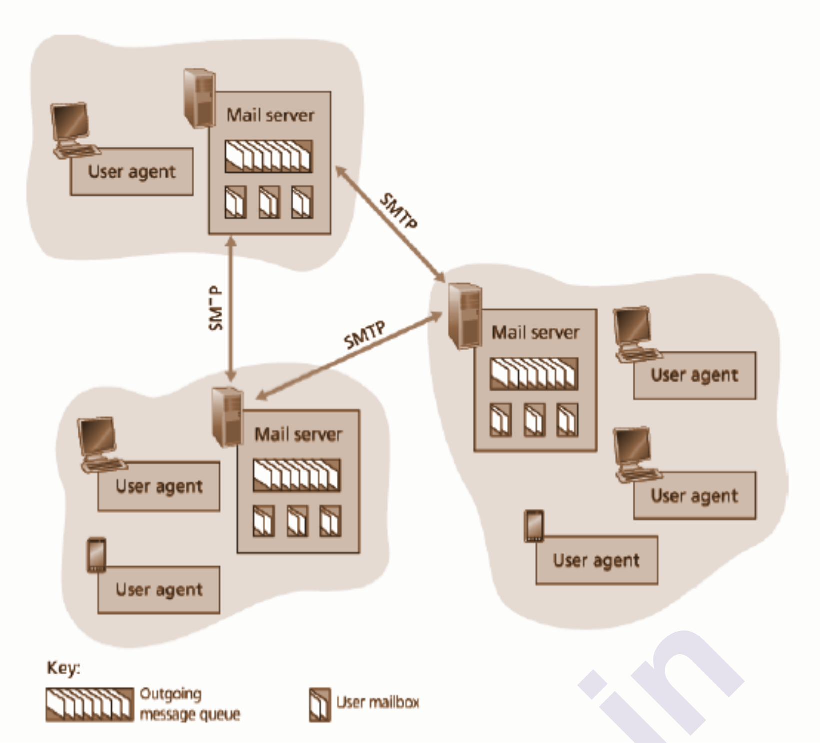


Fig. 9 a High-Level view of the internet E-Mail System

1.7.1 SMTP

SMTP is the heart of internet electronic mail. SMTP function is to move messages from senders' mail servers to the recipients' mail servers. The origin of SMTP is in 1982, SMTP is much older than HTTP. SMTP comes before HTTP. SMTP restricts the body of all mail messages to simple 7-bit ASCII code. When transmission capacity was scarce and no one was emailing large attachments or large image, audio, or video files. But nowadays, in the multimedia era, before being sent over SMTP the 7-bit ASCII restriction is a bit. It requires binary multimedia data to be encoded to ASCII format and it requires the corresponding ASCII message to be decoded back to binary after SMTP transport. HTTP does not require multimedia data to be ASCII encoded before transmitting.

To illustrate the basic operation of an SMTP server, let's walk through a common example: suppose Amit wants to send shyam a simple ASCII message.

1. Amit invokes his user agent for e-mail, provides shyam's e-mail address (for example, shyam@someschool.edu), composes a message, and instructs the user agent to send the message.
2. amit's user agent sends the message to his mail server, whose it is placed in a message queue.
3. The client side of SMTP, running on amit's mail server, finds the message in the message queue. it opens a TCP connection to an SMTP server, running on shyam's mail server.

4. After some initial SMTP handshaking, the SMTP client sends Amit's message into the TCP connection.
5. At shyam's mail server, the server side of SMTP receives the message. shyam's mail server then places the message in shyam's mailbox.
6. Shyam calls on his user agent to read the message at his convenience.

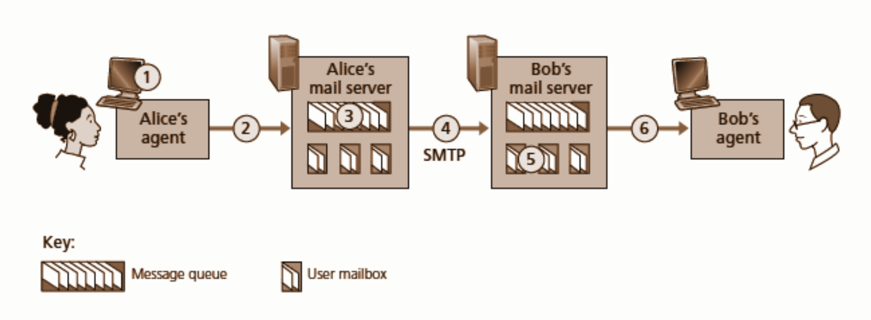


fig. 10Amit sends a message to Shyam

1.7.2 Comparison with HTTP (SMTP with HTTP)

1. HTTP & SMTP protocols are used to transfer files from one host to another host
2. **HTTP** transfers files from a web server to a web client.
3. **SMTP** transfers e-mail messages from one mail server to another mail server.
4. In SMTP & HTTP, when transferring the files, both persistent **HTTP** and **SMTP** use persistent connections. Both two protocols have common characteristics.
5. **HTTP** is mainly a pull protocol whose someone loads information on a web server and users use HTTP to pull the information from the server at their benefit.
6. **SMTP** is primarily a push protocol; the sending mail server pushes the file to the receiving mail server. On the other hand the TCP connection is initiated by the machine that wants to send the file.
7. **SMTP** requires each message, including the body of each message, to be in 7-bit ASCII format. If the message contains characters that are not 7-bit ASCII then the message has to be encoded into 7-bit ASCII. HTTP data does not impose this constraint.
8. Internet mail places all of this message's objects into one message.

1.7.3 Mail message formats

When Amit writes a simple mail or letter to shyam, he may include all kinds of peripheral header information at the top of this letter, like shyam's address, his own return address, and the date. Similarly, when an email message is sent from one person to another, a header containing peripheral information precedes the body of the message itself. This information is contained in a series of header lines, which are defined in RFC 5322. the header lines and the body of the message are divided by a blank line. RFC 5322 specifies the exact format for mail header lines as well as their semantic explanation. As with HTTP, each header line contains human readable text, consisting of a keyword followed by a: followed by a value. Some of the keywords are required and others are optional.

Each and every header must have a from: header line and a to: header line. a header may include a Subject: header line as well as others optional header lines. it is important to note that these header lines are different from the SMTP commands, the commands were part of the SMTP handshaking protocol. The header lines checked in this section are part of the mail message itself. a typical message header shown below:

from: amit@crepes.fr
to: shyam@hamburger.edu
subject: searching for the meaning of life.

After the message header, a blank line follows, and then the message body follows. It must use telnet to send a message to a mail server that contains some header lines, including the subject: header line.

1.7.4 Mail access protocols

Once SMTP server delivers the message from Amit's mail server to Shyam's mail server, the message is placed in Shyam's mailbox. Then shyam reads his mail by logging onto the server host and then executing a mail reader that runs on that host. In the early 1990s this was the standard way of doing things. But today, mail access uses client-server architecture. The typical user reads email with a client that executes on the user's end system, for example, on an office pc, a laptop, or a smartphone. By executing a mail client on a local pc, users enjoy a rich set of features, including the ability to view multimedia messages and attachments

1.8 DOMAIN NAME SYSTEM

Human beings can be identified in many ways. For example, we can be identified by the names that appear on our birth certificates or ID card and much more. Humans can be identified by our driver's license numbers. Even though each of these identifiers can be used to identify people, within a given context one identifier may be more appropriate than another. For e.g., the computers at the IRS prefer to use fixed-length social security numbers rather than birth certificate names. On the other side, ordinary people prefer the more mnemonic birth certificate names rather than social security numbers.

Internet hosts can be identified in many ways like humans. One identifier for a host is its hostname. Hostnames such as ibn.com, www.yahoo.com, somaiya.edu, and mu.ac.in are mnemonic and are therefore appreciated by humans. However, hostnames provide little, if any, information about the location within the internet of the host. (A hostname such as www.mu.ac.in, which ends with the country code .in, tells us that the host is probably in India, but doesn't say much more.) Furthermore, because hostnames can consist of variable length alphanumeric characters, they would be difficult to process by routers. For these reasons, hosts are also identified by IP addresses.

An IP address consists of four bytes and has a rigid hierarchical structure. An IP address looks like 192.168.3.1, which's each period separates one of the bytes expressed in decimal notation from 0 to 255. An IP address is hierarchical because this is scanning the address from left to right, obtaining more and more specific information about who's the host is located on the internet. Similarly, when we scan a postal address from bottom to top, this obtains more and more specific information about who's the addressee is located.

1.8.1 Services provided by DNS

There are two ways to identify a host by a hostname and by an IP address. People prefer the more mnemonic hostname identifier, while routers prefer fixed-length, hierarchically structured IP addresses. In order to reconcile these preferences, we need a directory service that translates hostnames to IP addresses. This is the main task of the internet's domain name system (DNS). The DNS is

1. A distributed database implemented in a hierarchy of DNS servers, and
2. An application-layer protocol that allows hosts to query the distributed database.

The DNS protocol runs over UDP and uses PORT 53.

DNS is commonly employed by these application layer protocols including HTTP, SMTP, and FTP to translate user supplied hostnames to IP addresses. As an example, consider what happens when a browser (that is, an HTTP client), running on some user's host, requests the url www.somaiya.edu/index.html. in order for the user's host to be able to send an HTTP request message to the web server www.somaiya.edu, the user's host must first obtain the IP address of www.somaiya.edu.

This is done as follows.

1. The same user machine runs the client side of the DNS application.
2. The browser extracts the hostname, www.somaiya.edu, from the URL and passes the hostname to the client side of the DNS application.

3. The DNS client sends a query containing the hostname to a DNS server.
4. The DNS client eventually receives a reply, which includes the IP address for the hostname.
5. Once the browser receives the IP address from DNS, it can initiate a TCP connection to the HTTP server process located at PORT 80 at that IP address.

DNS provides a few other important services in addition to translating hostnames to IP addresses:

a. **Host aliasing**. a host with a cumbersome hostname can have one or more alias names. For example, a hostname like relay1.west-coast.enterprise.com could have, say, two aliases such as enterprise.com and www.enterprise.com. in this case, the hostname relay 1.west coast. Enterprise.com is said to be an authorized hostname. Alias hostnames, when available, are often more mnemonic than authorized hostnames. DNS can be requested by an application to obtain the authorized hostname for a supplied alias hostname as well as the IP address of the host.

b. **Mail server aliasing**. It is highly advisable that e-mail addresses be mnemonic. For example, if shyam has an account with hotmail, shyam's e-mail address might be as simple as shyam@hotmail.com. However, the hostname of the hotmail mail server is more complicated and much less mnemonic than simply hotmail.com, for example, the authorized hostname might be something like relay1.west-coast.hotmail.com. DNS can be invoked by a mail application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.

c. **Load distribution**. DNS is also used to perform load distribution among replicated servers, such as replicated web servers. Busy sites, such as ibn.com, are replicated over multiple servers, with each server running on a different end system and each having a different IP address. For replicated web servers, a set of IP addresses is thus connected with one canonical hostname. The DNS database contains this set of IP addresses. When clients make a DNS query for a name mapped to a set of addresses, the server responds with the entire set of IP addresses, but rotates the ordering of the addresses within each reply. Because a client typically sends its HTTP request message to the IP address that is listed first in the set, DNS rotation distributes the traffic among the different copies of servers. DNS rotation is also used for email so that multiple mail servers can have the same alias name.

1.8.2 Overview of how DNS works

It presents a high-level overview of how DNS works. Suppose that some application running in a user's host needs to translate a hostname to an IP address. The application will invoke the client side of DNS, specifying the hostname that needs to be translated. DNS in the user's host then takes over, sending a query message into the network. All DNS query and reply messages are sent within UDP datagrams to PORT 53. After a delay, ranging from milliseconds to seconds, DNS in the user's host receives a DNS reply message that provides the desired mapping. This mapping is then passed to the invoking application. Thus, from the perspective of the invoking application in the user's host, DNS is a black box providing a simple, straightforward translation service. but in fact, the black box that implements the service is complex, consisting of a large number of DNS servers distributed around the globe, as well as an application-layer protocol that specifies how the DNS servers and querying hosts communicate.

A simple design for DNS would have one DNS server that contains all the mappings. In this centralized design, clients simply direct all queries to the single DNS server, and the DNS server responds directly to the querying clients. Although the simplicity of this design is attractive, it is inappropriate for today's internet, with its vast (and growing) number of hosts. The problems with a centralized design include:

a. **A single point of failure.** If the DNS server crashes, so does the entire internet!

b. **Traffic volume**. A single DNS server would have to handle all DNS queries.

c. **Distant centralized database**. A single DNS server cannot be "close to" all the querying clients. If we put the single DNS server in Mumbai city, then all queries from Australia must travel to the other side of the globe, maybe over slow and congested links. This can lead to significant delays.

d. **Maintenance.** The single DNS server would have to keep records for all internet hosts. Not only would this centralized database be large, but it would have to be updated frequently to account for every new host.

1.8.3 DNS Records and Messages

The DNS servers that together implement the DNS distributed database store resource records (RRS); including RRS that provide hostname-to-IP address mappings. Each DNS reply message carries one or more resource records.

A resource record is a four-tuple that contains the following fields:

(Name, Value, Type, TTL)

TTL is the time to live off the resource record; it determines when a resource should be removed from a cache. In the example records given below, we ignore the TTL field. The meaning of name and value depend on type:

- If Type=A, then name is a hostname and value is the IP address for the hostname. Thus, a type record provides the standard hostname-to-IP address mapping. As an example, (relay1.bar.foo.com, 145.37.93.126, a) is a type of record.
- If Type=NS, then name is a domain (such as foo.com) and value is the hostname of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the domain. This record is used to route DNS queries further beside in the query chain. As an example, (foo.com, DNS.foo.com, NS) is a type NS record.
- If Type=CNAME, then value is a valid or canonical hostname for the alias hostname name. This record can provide querying hosts the canonical name for a hostname. As an example, (foo.com, relay1.bar.foo.com, CNAME) is a CNAME record.
- If Type=MX, the value is the valid or canonical name of a mail server that has an alias hostname name. As an example, (abc.com, mail.bar.abc.com, MX) is an MX record. MX records allow the hostnames of mail servers to have simple aliases. Note that by using the MX record, a company can have the same aliased name for its mail server and for one of its other servers (such as its web server). To obtain the canonical name for the mail server, a DNS client would query for an MX record; to obtain the canonical name for this server, the DNS client would query for the CNAME record.

DNS messages

There are only two types of DNS messages. Both query and reply messages have the same format, as shown in figure 3.the semantics of the various fields in a DNS message are as follows:

a. The first 12 bytes is the header section of message, which has a number of fields. In the message the first field is a 16-bit number that identifies the query. This identifier is copied into the reply message to a query, allowing the client to match received replies with sent queries. These are a number of flags in the flag field. A 1-bit reply flag indicates, the message is a query (0) or a reply (1). A 1-bit recursion-desired flag is set when a client (host or DNS server) capable that the DNS server perform recursion when it doesn't have the record. A 1-bit recursion available field is set in a reply if the DNS server supports recursion. In the header, those are also four number-of fields. Which indicate the number of occurrences of the four types of data parts that follow the header.

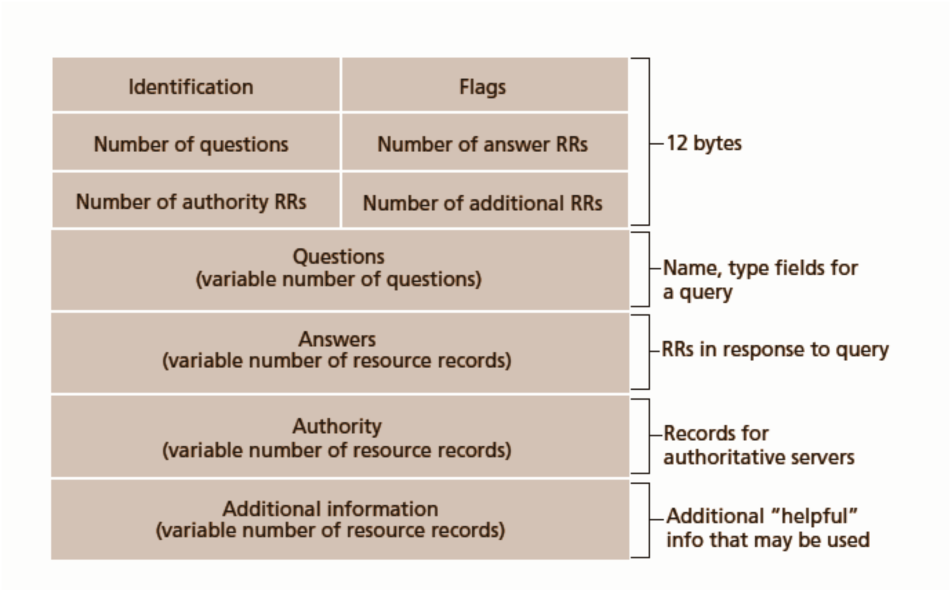

Identification
Flags
Number of questions
Number of answer RRs
Number of authority RRs
Number of additional RRs
12 bytes
Questions
(variable number of questions)
Name, type fields for a query
Answers
(variable number of resource records)
RRs in response to query
Authority
(variable number of resource records)
Records for authoritative servers
Additional information
(variable number of resource records)
Additional "helpful" info that may be used

Fig. 11 DNS message format

b. The question part of the message contains information about the query that is being made. This part of message includes a name field that contains the name that is being queried, and a type field that indicates the type of question being asked about the name for example, a host address associated with a name (Type A) or the mail server for a name (Type MX).

c. In a reply from a DNS server, the answer part of the message contains the resource records for the name that was originally queried. Recollect that in each resource record those is the type (for example, A, NS, CNAME, and MX), the value, and the TTL. A reply can return multiple RRS in the answer part of the message, since a hostname can have multiple IP addresses.

d. The authority section of the message contains records of other authoritative servers.

e. The additional section of the message contains other helpful records. For example, the answer field in reply to an MX query contains a resource record providing the canonical hostname of a mail server.

1.7 SUMMARY

1. In this chapter, we've studied the conceptual and the implementation aspects of network applications.

2. We've learned about the ubiquitous client-server architecture adopted by many Internet applications and seen its use in the HTTP, FTP protocols.

3. We've studied these important application-level protocols, and their corresponding associated applications (the web, file transfer) in some detail.

4. We have also learn difference between Non-Persistent and Persistent Connections
5. We have seen the HTTP Message format & HTTP Response Message.

1.8 REFERENCE FOR FURTHER READING

1. A Computer Networking, A Top-Down Approach Kurose & ross
2. TCP/IP Protocol Suite 4 edition, Beerhouse Frozen, McGraw-Hill Science

1.9 UNIT END EXERCISES

1. What is meant by a handshaking protocol?
2. Why is it said that FTP sends control information “out-of-band”?
3. Explain the Protocol layer and their services?
4. Write a short note on:
   a. Non-Persistent and Persistent Connections
   b. HTTP Message Format
   c. HTTP Response Message
   d. User-Server Interaction: Cookies

❖❖❖❖

2

NETWORKING - II

Unit Structure

2.1 OBJECTIVE

1. To understand the major components of electronic mail in the internet
2. To understand the services provided by DNS
3. To learn the mail message formats & access protocols.
4. To understand the transport-layer services

2.2 TRANSPORT-LAYER SERVICES

A transport-layer protocol provides for rational communication between application processes running on different hosts. In logical communication, we mean that from an application's viewpoint, it is as if the hosts running the processes were directly connected. In the real world, the hosts may be on opposite sides of the planet, connected via numerous routers and a wide range of link types. The application processes use the logical communication provided by the transport layer to send messages to each other, free from the worry of the details of the physical infrastructure used to carry these messages. Figure 4 shows the notion of logical communication.

As shown in figure 4, function of transport layer

1. Transport-layer protocols are implemented in the end systems.
2. The transport layer converts the application-layer messages it receives from a sending application process into transport-layer packets, known as transport-layer segments in internet terminology.
3. The transport layer then passes the segment to the network layer at the sending end system, whose segment is encapsulated within a network-layer packet (a datagram) and sent to the destination.
4. The network routers act only on the network-layer fields of the datagram.
5. At the receiving side, the network layer extracts the transport-layer segment from the datagram and passes the segment up to the transport layer. the transport layer then processes the received segment, making the data in the segment available to the receiving application.

2.2.1 Relationship between Transport and Network layers

The transport layer is just above the network layer in the protocol stack. Whereas a transport-layer protocol provides logical communication between processes running on different hosts, a network-layer protocol provides logical communication between hosts.

2.2.2 Overview of the Transport Layer in the Internet

In TCP/IP, the two distinct transport-layer protocols available to the application layer. Among two one of these protocols is UDP (user datagram protocol), which provides an unreliable, connectionless service to the invoking application. The second protocol is TCP (transmission control protocol), which provides a reliable, connection-oriented service to the invoking application. When designing a network application, the application developer selects between UDP and TCP when creating sockets.

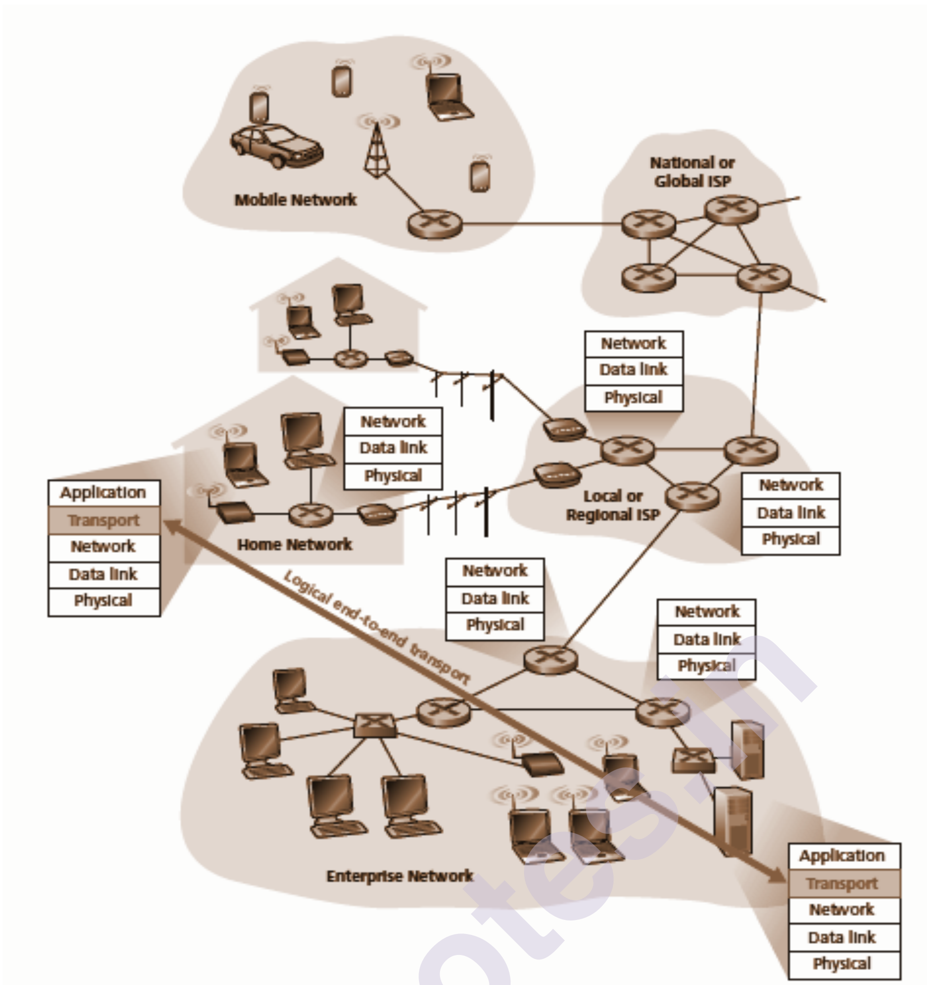


Fig. 4 the transport layer provides logical rather than physical communication between application processes

2.3 MULTIPLEXING AND DEMULTIPLEXING

In the transport-layer protocol multiplexing and demultiplexing, is extending the host-to-host delivery service provided by the network layer to a process-to-process delivery service for applications running on the hosts. A multiplexing & demultiplexing service is needed for all computer networks. at the destination host, the transport layer receives segments from the network layer. the transport layer has the responsibility of delivering the data in these segments to the appropriate application process running on the host. Example. User sitting in front of the computer and user are downloading web pages while running one FTP session and two telnet sessions. These four network application processes run two telnet processes, one FTP process, and one HTTP process. When the transport layer in the user computer receives data from the network layer below, it needs to direct the received data to one of these four processes.

In the network application, a process can have one or more sockets, doors through which data passes from the network to the process and through which data passes from the process to the network back. Thus, as shown in figure 4, the transport layer in the receiving host does not actually deliver data directly to a process. Each socket has a unique identifier. The format of the identifier depends on whether the socket is a UDP or a TCP socket.

Delivering the data in a transport-layer segment to the correct socket is called demultiplexing. The job of gathering data chunks at the source host from different sockets, encapsulating each data chunk with header information to create segments, and passing the segments to the network layer is called multiplexing. The transport layer in the middle host must also gather outgoing data from these sockets, form transport-layer segments, and pass these segments down to the network layer.

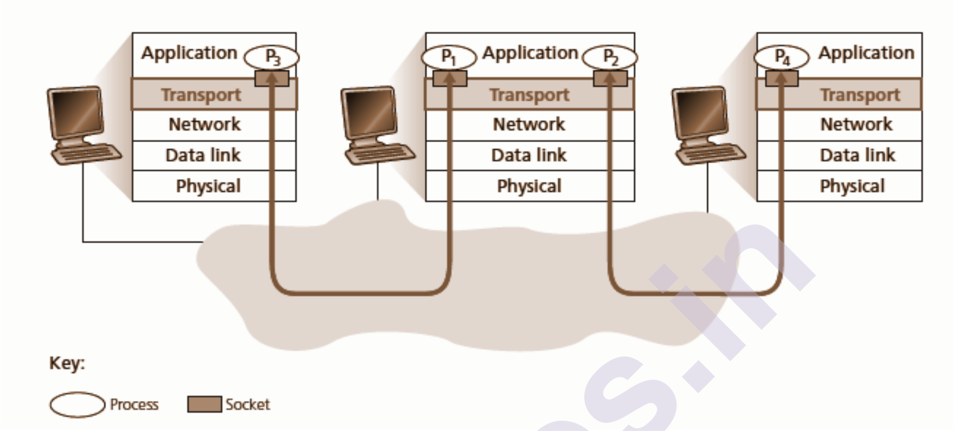


Fig. 5 transport-layer multiplexing and demultiplexing

The roles of transport-layer multiplexing and demultiplexing:

The sockets have unique identifiers, and that each segment has special fields that indicate the socket to which the segment is to be delivered. These special fields, Shown in figure 6, are the source port number field and the destination port number field, The 16-bit port number, between 0 to 65535. The port numbers ranging from 0 to 1023 are called well-known port numbers and are restricted, which means that they are reserved for use by well-known application protocols such as HTTP and FTP.

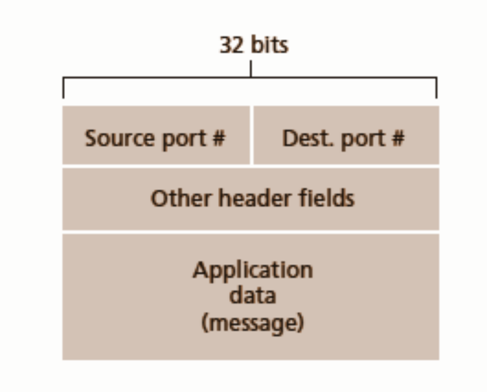


Fig. 6 Source and destination port-number fields in a transport-layer segment

Connectionless Multiplexing and Demultiplexing

The program running in a host can create a UDP socket with the line

```
clientsocket = socket(socket.af_inet, socket.sock_dgram)
```

When a UDP socket is created in this way, the transport layer automatically assigns a port number to the socket. The transport layer allots a port number in the range 1024 to 65535 that is currently not being used by any other UDP port in the host. In python program after creating the socket to associate a specific port number (say, 19157) to this UDP socket via the socket bind() method:

```
clientsocket.bind((‘’, 19157))
```

Connection-Oriented multiplexing and demultiplexing

One best difference between a TCP socket and a UDP socket is that a TCP socket is identified by a four-tuple:

1. sourceip address.
2. source port number.
3. destinationip address.
4. destination port number.

when a TCP segment arrives from the network to a host, the host uses all four values to direct the segment to the appropriate socket. In contrast with udp, two arriving TCP segments with different source IP addresses or source port numbers will be directed to two different sockets. to gain further insight, let’s rethink the TCP client-server programming.

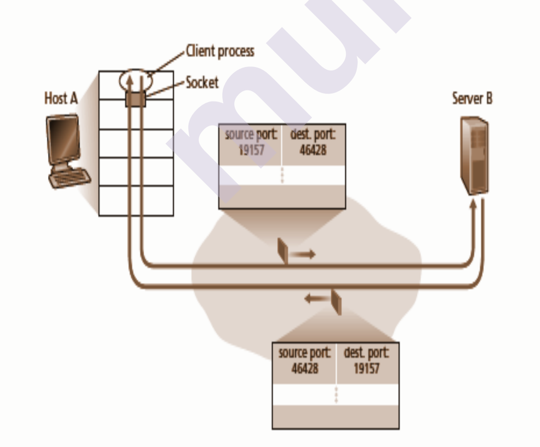


Fig. 7 The inversion of source and destination port numbers

- ❑ The TCP server application has a “welcome socket,” that waits for connection establishment requests from TCP clients on port number 12000.
- ❑ The TCP client creates a socket and sends a connection establishment request message with the lines:

```
clientsocket = socket(af_inet, sock_stream)
clientsocket.connect((servername,12000))
```

- A connection initiation request is nothing more than a TCP segment with destination port number 12000 and a special connection-establishment bit set in the TCP header. The message also includes a source port number that was chosen by the user.
- The host OS of the computer running the server process receives the incoming connection-request segment with destination port 12000, it locates the server process that is waiting to accept a connection on port number 12000. the server process then creates a new socket:

```
connectionsocket, addr = serversocket.accept()
```

- The transport layer at the server notes the following four values in the connection-request segment:
  a. the source port number in the segment
  b. the ip address of the source host
  c. the destination port number in the segment
  d. its own ip address.

The recently created connection socket is identified by these four values; all subsequently arriving segments whose source port, source IP address, destination port, and destination IP address match these four values will be demultiplexed to this socket. with the TCP connection now in place, the client and server can now send data to each other.

2.4 UDP

Figure 8 shows the relationship of the User Datagram Protocol (UDP) to the other protocols and layers of the TCP/IP protocol suite: UDP is located between the application layer and the IP layer, and serves as the intermediary between the application programs and the network operations.

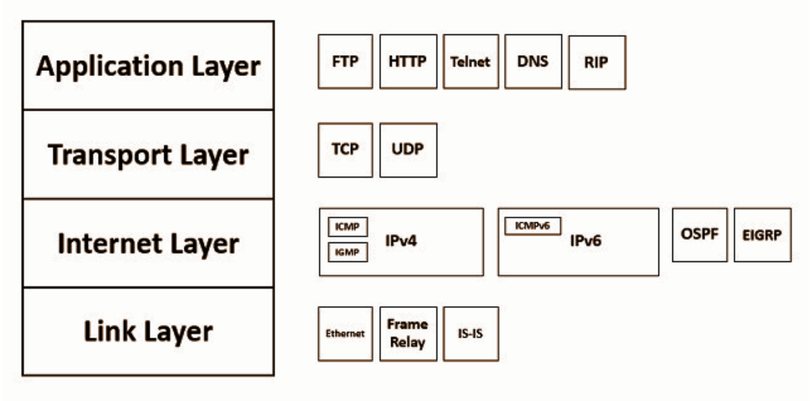


Fig. 8 Position of UDP in the TCP/IP protocol suite

A transport layer protocol usually has several responsibilities. One is to create a process-to-process communication. UDP uses port numbers.

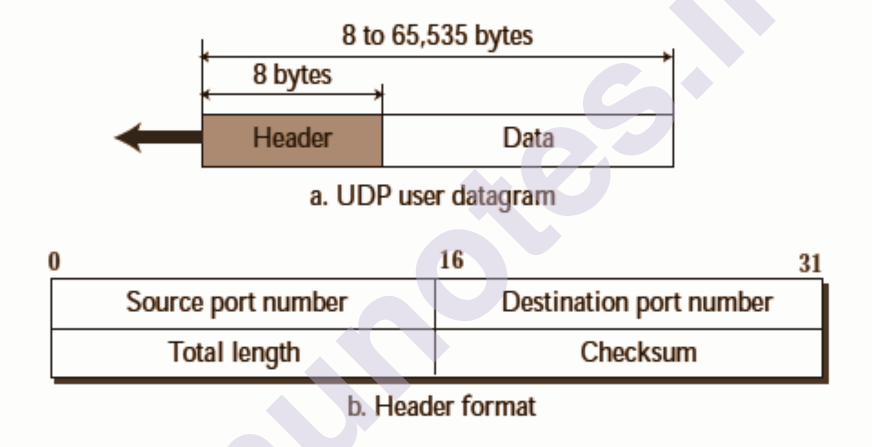
Second responsibility is to provide control mechanisms at the transport level. UDP doesn't have a flow control mechanism and no acknowledgement for received packets. UDP does provide error control to some extent. If UDP detects an error in the received packet, it drops it.

UDP is a connectionless, unreliable transport protocol. UDP provides process-to-process communication instead of host-to-host communication.

UDP is a very simple protocol using a minimum of cost. If a process wants to send a small segment and does not look much about reliability, it can use UDP. Sending a small segment using UDP takes much less interaction between the sender and receiver than using TCP.

2.4.1 USER DATAGRAM

UDP packets, called user datagrams. UDP has a fixed-size header of 8 bytes. Figure 9 shows the format of a user datagram in detail.



a. UDP user datagram

Source port number	Destination port number
Total length	Checksum

b. Header format

Fig. 9 User datagram format

- **Source port number.**
  1. This is the port number used by the process running on the source host.
  2. It is 16 bits long, port number can range from 0 to 65,535.
  3. If the source host is the client, the port number, in most cases, is a transitory. port number requested by the process.
  4. If the source host is the server, the port number is a well-known port number.

- **Destination port number.**
  1. This is the port number, running on the destination host.
  2. 16 bits long.
  3. If the destination host is the server, is a well-known port number.
  4. If the destination host is the client, is a transitory port number.

- **Length.**

  1. Length is a 16-bit field that defines the total length of the user datagram, header plus data.
  2. It defines a total length of 0 to 65,535 bytes.
  3. A user datagram is enclosed in an IP datagram.
  4. There is a field in the IP datagram that defines the total length.
  5. There is another field in the IP datagram that defines the length of the header.

UDP datagram that is encapsulated in an IP datagram.

UDP length= IP length− IP header's length

- **Checksum.** This field is used to detect errors over the entire user datagram (header
  plus data). The checksum is discussed in the next section.

  Example
  The following is a dump of a UDP header in hexadecimal format.

  CB84000D001C001C

a. The source port number is the first four hexadecimal digits (CB8416), which means that the source port number is 52100.

b. The destination port number is the second four hexadecimal digits (000D16), which means that the destination port number is 13.

c. The third four hexadecimal digits (001C16) define the length of the whole UDP packet as 28 bytes.

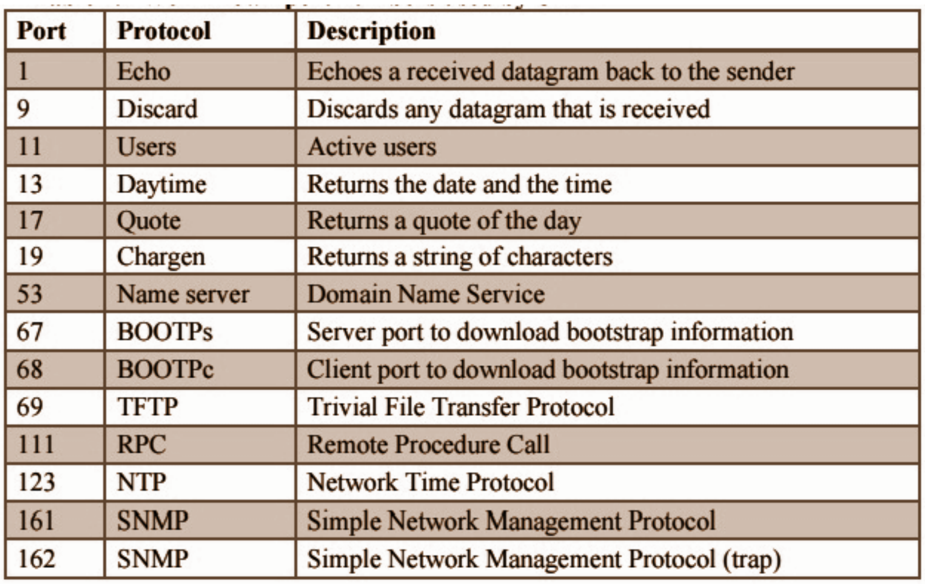
d. The length of the data is the length of the whole packet minus the length of the header, or 28 – 8 = 20 bytes.

e. Since the destination port number is 13, the packet is from the client to the server.

f. The client process is the Daytime.

2.4.2 UDP SERVICES

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 1



Port	Protocol	Description
1	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	Name server	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

Table 1. Well-known Ports used with UDP

Connectionless Services

UDP provides a connectionless service. The user datagram sent by UDP is an independent datagram. There is no connection between the different user datagrams even if they are coming from the same source process and going to the same destination program. The user datagrams are not numerical. There is no connection establishment and no connection termination as is the case for TCP. This means that each user datagram can travel on a different path.

Flow Control

UDP is a very simple protocol. There is no flow control, and hence no window mechanism. The receiver may overflow with incoming messages. The lack of flow control means that the process using UDP should provide for this service, if needed.

Error Control

There is no error control mechanism in UDP except for the checksum. The sender does not realize that if a message has been lost or duplicated. When the receiver found an error through the checksum, the user datagram is discarded.

Checksum

UDP checksum calculation is different from the one for IP. Here the checksum includes three sections:

1. A pseudo header,
2. UDP header, and
3. Data coming from the application layer

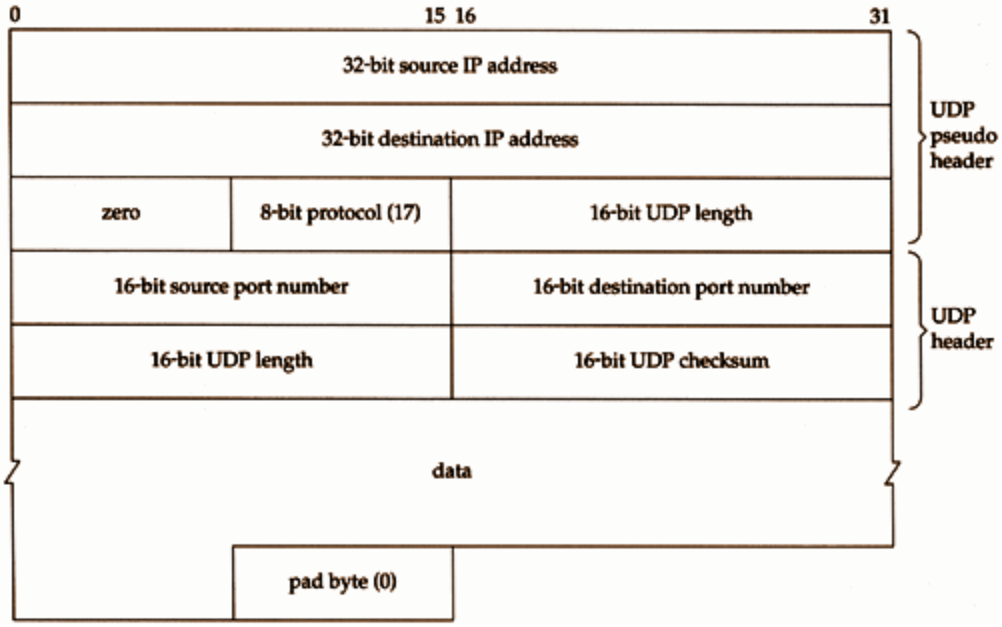


Fig. 10 UDP Header

If the checksum does not include the pseudo header, a user datagram may arrive safe and sound. However, if the IP header is corrupted, it may be delivered to the wrong Host. The protocol field is added to make sure that the packet belongs to UDP, and not to TCP. The value of the protocol field in the datagram for UDP is 17. If this value is changed during transmission, the checksum calculation at the receiver will detect it and UDP drops the packet. It is not delivered to the wrong protocol.

Congestion Control

UDP is a connectionless protocol; it does not provide congestion control. UDP assumes that the packets sent are small and sporadic, and cannot create congestion in the network.

Encapsulation and Decapsulation

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

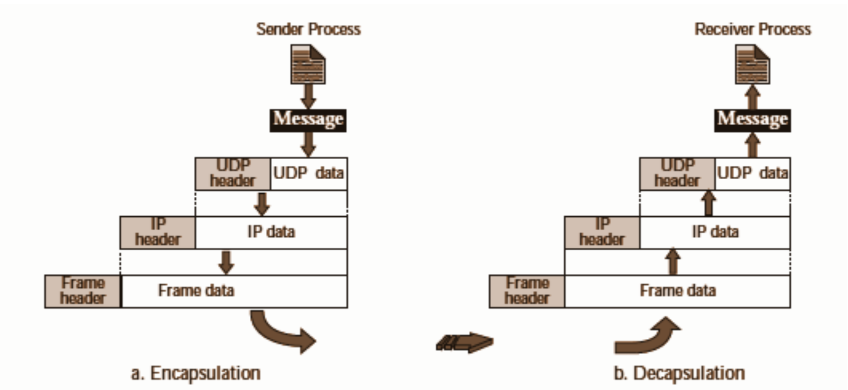


Fig. 11Encapsulation and decapsulation

Encapsulation

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

Decapsulation

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

Queuing

In UDP, queues are associated with ports At the client site, when a process starts, it requests a port number from the operating system. Some implementations create both an incoming and an outgoing queue associated with each process. Other implementations create only an incoming queue associated with each process

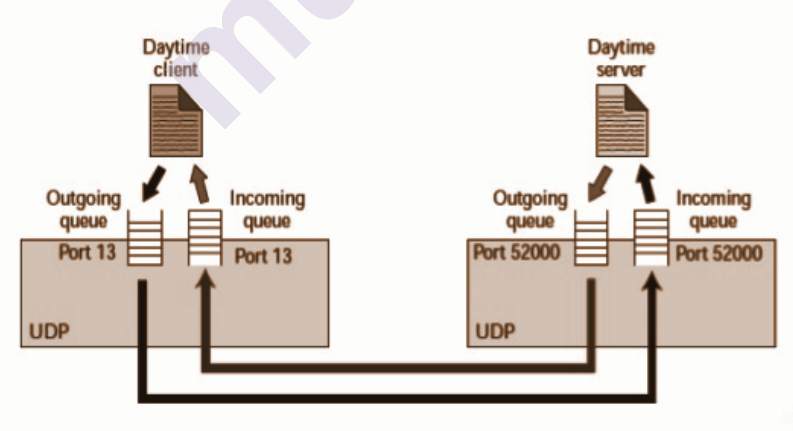


Fig. 12 Queues in UDP

2.4.3 UDP APPLICATIONS

- UDP is suitable for a process that requires simple request-response communication with flow and error control mechanisms. It is not usually used for a process such as FTP that needs to send bulk data.

- UDP is suitable for a process with internal flow and error-control mechanisms.
- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- UDP is used for management processes such as SNMP.
- UDP is used for some route updating protocols such as Routing Information Protocol (RIP).
- UDP is normally used for real-time applications that cannot tolerate uneven delay between sections of a received message.

2.5 TCP

Figure 13 shows the relationship of TCP to the other protocols in the TCP/IP protocol suite. TCP lies between the application layer and the network layer, and serves as the intermediary between the application programs and the network operations.

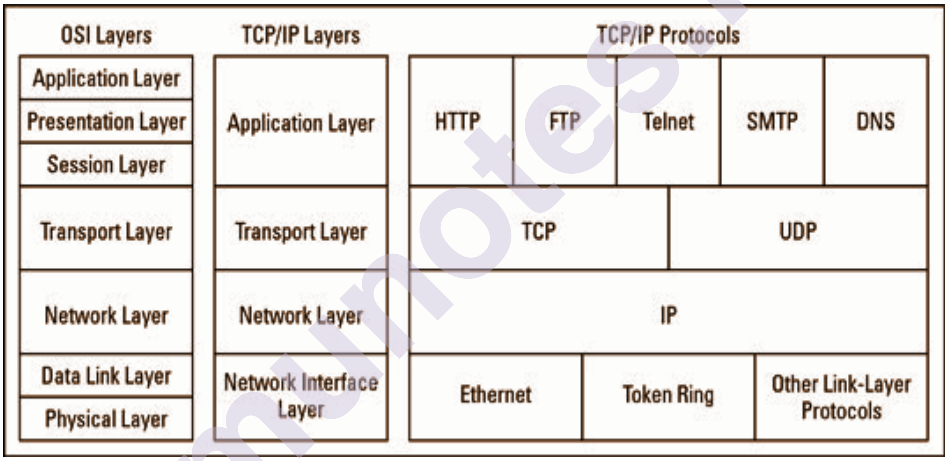
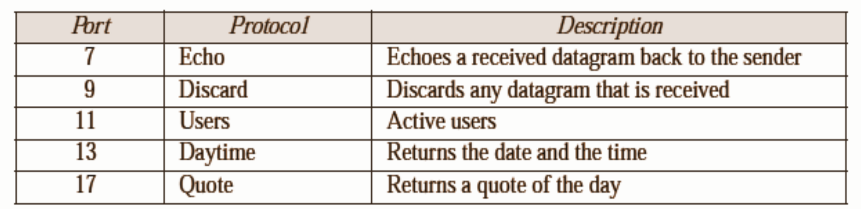


Fig. 13 TCP/IP protocol suite

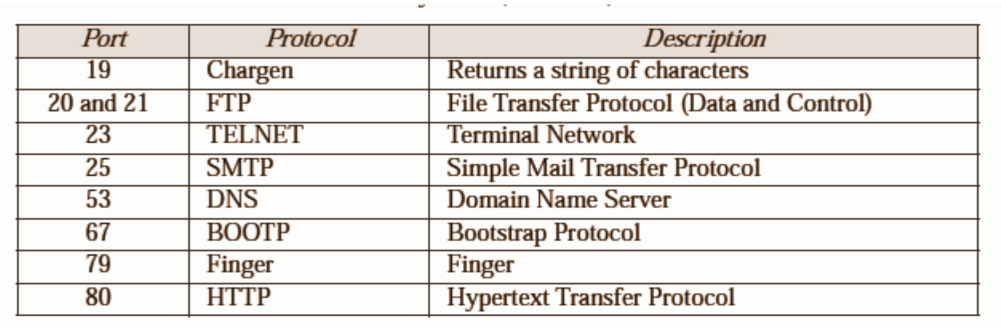
Process-to-Process Communication

As with UDP, TCP provides process-to-process communication using port numbers.
Table 2 & 3 lists some well-known port numbers used by TCP.



Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day

Table 2. Well-known Ports used by TCP



Port	Protocol	Description
19	Chargen	Returns a string of characters
20 and 21	FTP	File Transfer Protocol (Data and Control)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol

Table 3. Well-known Ports used by TCP

Stream Delivery Service

TCP, unlike UDP, is a stream-oriented protocol. In UDP, a process sends messages with predefineborder to UDP for delivery. UDP adds its own header to each of these messages and delivers to the destination or to IP for transmission. Each message from this process is called a user datagram.

TCP, permits 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 adomain 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 14. The sending process writes to the stream of bytes and the receiving process reads from them.

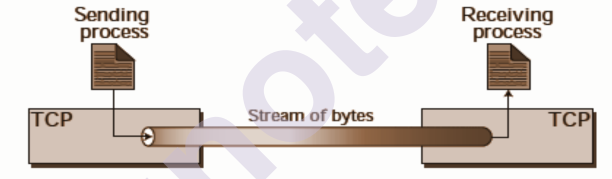


Fig. 14.Stream delivery

Sending and Receiving Buffers

The sending and the receiving processes may not definitely write or read data at the same rate, TCP required buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction. On sided to implement a buffer is to use a circular array of 1-byte locations as shown in Figure 15shows two buffers of 20 bytes each.

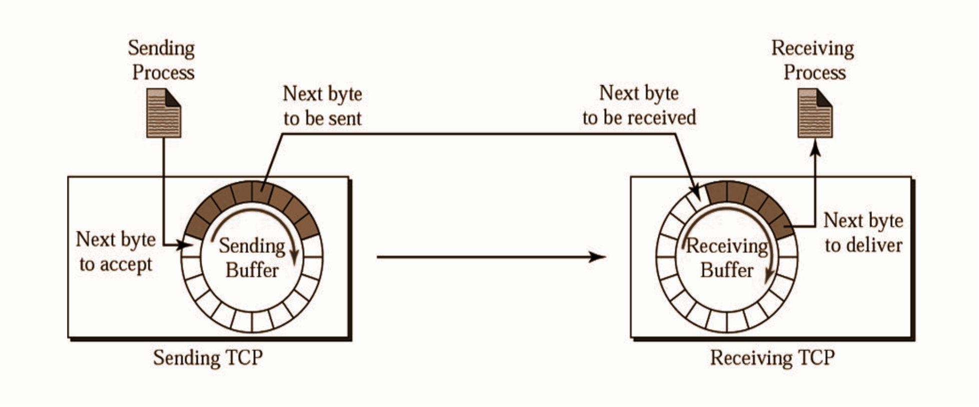


Fig. 15 Sending and receiving buffers

Segments

At the transport layer, TCP groups a number of bytes together into a packet called a segment. TCP adds a header to each segment 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. Figure 16 shows how segments are created from the bytes in the buffers.

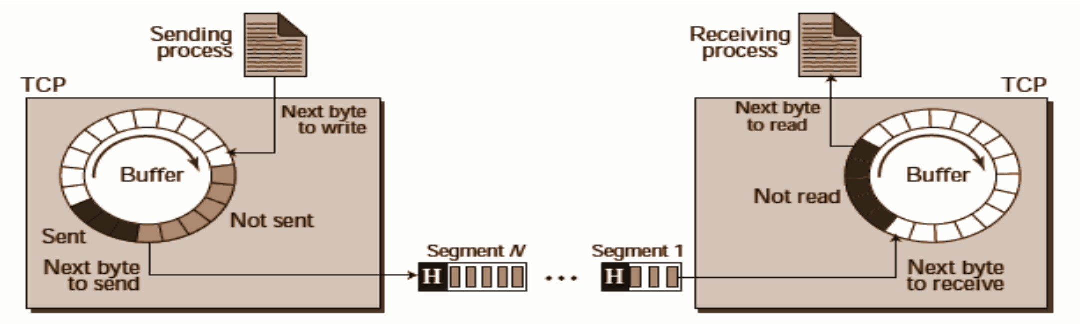


Fig. 16 TCP segments

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. 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 is exchanged in both directions.
3. The connection is terminated.

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.

2.5.1 TCP FEATURES

Numbering System

TCP software keeps track of the segments being transmitted or received, there is no field for a segment number value in the segment header. Instead, there are two fields called the sequence number and the acknowledgment number. These two fields refer to a byte number.

Byte Number

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with an arbitrarily generated number.

Sequence Number

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

Acknowledgment Number

When a connection is established, both parties can send and receive data at the same time. Each party numbers the bytes, usually with a different starting byte number. The sequence number in each direction shows the number of the first byte carried by the segment. Each party also uses an acknowledgment number to confirm the bytes it has received.

The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative.

Flow Control

UDP provides flow control. The sending TCP controls how much data can be accepted from the sending process; the receiving TCP controls how much data can be sent by the sending TCP. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte oriented flow control.

Error Control

To provide reliable service, TCP implements an error control mechanism. error control considers a segment as the unit of data for error detection, error control is byte-oriented.

Congestion Control

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

2.5.2 SEGMENT

A packet in TCP is called a segment.

Format

The format of a segment is shown in Figure 17. The segment consists of a header of 20 to 60 bytes, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

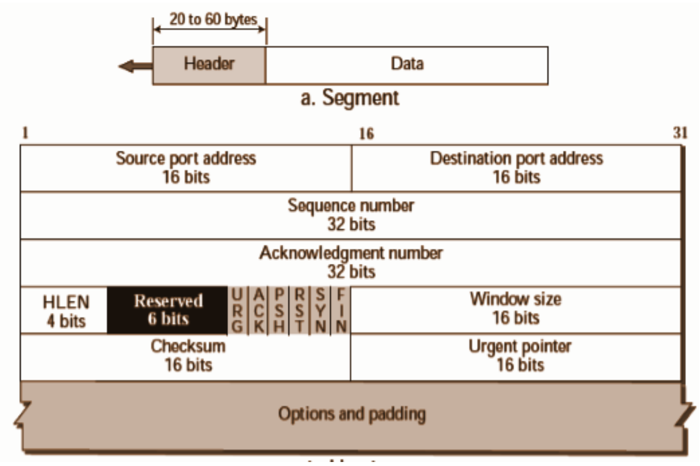


Fig. 17 TCP segment format

- **Source port address.** This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.
- **Destination port address**. This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.
- **Sequence number**. This 32-bit field defines the number assigned to the first byte of data contained in this segment.
- **Acknowledgment number.** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party.
- **Header length**. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 (5× 4= 20) and 15 (15× 4= 60).
- **Reserved.** This is a 6-bit field reserved for future use.
- **Control.** This field defines 6 different control bits or flags as shown in Figure 18 One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.

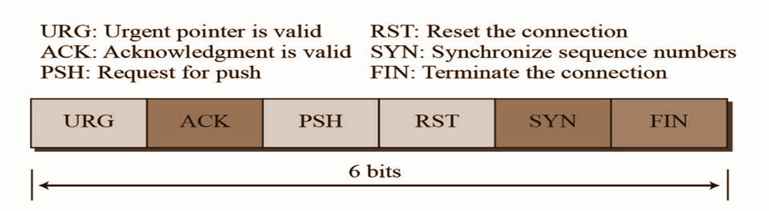

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection
URG
ACK
PSH
RST
SYN
FIN
6 bits

Fig. 18 Control field

- **Window size.** This field defines the window size of the sending TCP in bytes. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes.
- **Checksum.** This 16-bit field contains the checksum.

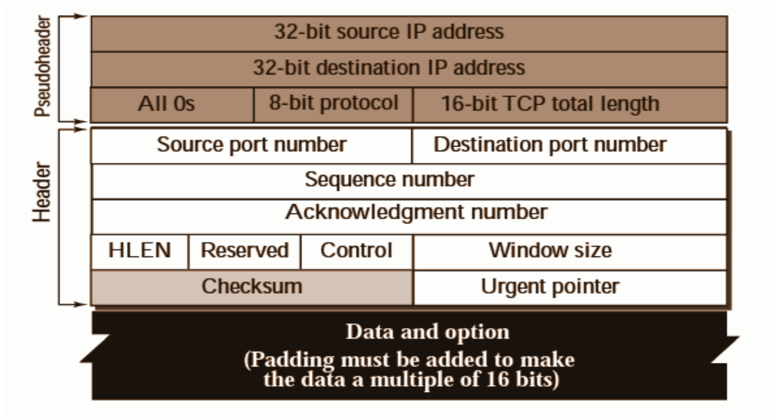


Fig. 19 Pseudoheader added to the TCP datagram

- **Urgent pointer.** This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data.
- **Options.** There can be up to 40 bytes of optional information in the TCP header.

2.6 TCP CONGESTION CONTROL

Congestion control in TCP is based on both open-loop and closed-loop mechanisms. TCP uses a congestion window and a congestion policy that avoids congestion and detects and alleviates congestion after it has occurred.

Congestion Window

If the network cannot deliver the data as fast as it is created by the sender, it must tell the sender to slow down. In other words, in addition to the receiver, the network is a second entity that determines the size of the sender's window.

The sender has two pieces of information: the receiver-advertised window size and the congestion window size. The actual size of the window is the minimum of these two.

Actual window size = minimum (rwnd, cwnd)

Congestion Policy

TCP's general policy for handling congestion is based on three phases: slow start, congestion avoidance, and congestion detection. In the slow start phase, the sender starts with a slow rate of transmission, but increases the rate rapidly to reach a threshold. When the threshold is reached, the rate of increase is reduced. Finally if ever congestion is detected, the sender goes back to the slow start or congestion avoidance phase, based on how the congestion is detected.

2.7 SUMMARY

1. We learned that a transport-layer protocol can provide reliable data transfer even if the underlying network layer is unreliable
2. We learned that TCP is complex, involving connection management, flow control, and round-trip time estimation, as well as reliable data transfer.
3. we took a close look at TCP, the Internet's connection-oriented and reliable transport-layer protocol.
4. We examined congestion control from a broad perspective, we showed how TCP implements congestion control.
5. we learned that TCP implements an end-to-end congestion-control mechanism.

2.8 REFERENCE FOR FURTHER READING

1. Computer Networking: A Top-Down Approach 6th edition, James F. Kurose, Keith W. Ross, Pearson (2012).
2. TCP_IP Protocol Suite 4th ed. - B. Forouzan (McGraw-Hill, 2010) BBS

2.9 UNIT END EXERCISES

1. Describe why an application developer might choose to run an application over UDP rather than TCP.
2. What is the difference between Multiplexing and Demultiplexing?
3. Explain Difference TCP & UDP.
4. What mechanism is used in TCP Congestion Control.
5. Explain & List the UDP Applications.

❖❖❖❖

3

NETWORKING - III

Unit Structure

3.1 OBJECTIVE

1. To learn exactly how the network layer implements the host-to-host communication service.
2. To examine two broad approaches towards structuring network-layer packet delivery the datagram and the virtual-circuit model—and see the fundamental role that addressing plays in delivering a packet to its destination host.
3. To understand an important distinction between the forwarding and routing functions of the network layer.

3.2 INTRODUCTION

For example a simple network with two hosts, X1 and X2, and several routers on the path between X1 and X2. Suppose that X1 is sending data to X2, and consider the role of the network layer in these hosts and in the intervening routers. The network layer in X1 takes messages from the transport layer in X1, encapsulates each message into a datagram and then sends the datagrams to its nearby router, R1. At the receiving host, X2, the network layer receives the datagrams from its nearby router R2, extracts the transport-layer segments, and delivers the segments up to the transport layer at X2. The primary role of the routers is to forward datagrams from input links to output links.

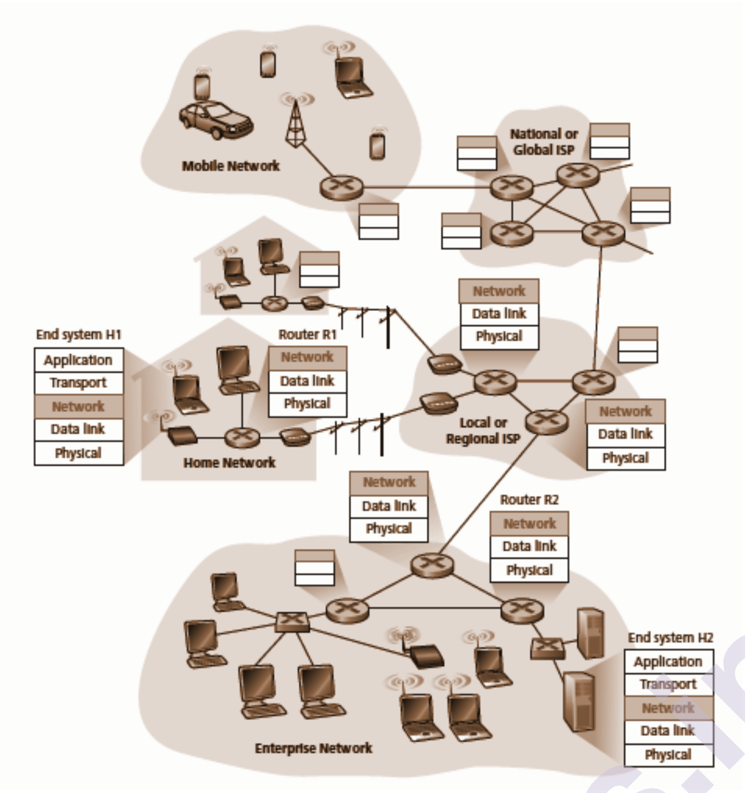


Fig. 1 Network Layer

Forwarding and Routing:

The role of the network layer is simple to move data packets from a sending host to a receiving host.

Two network-layer functions:

1. **Forwarding.** When a packet comes at a router's input link, the router must move the packet to the appropriate output link. For example, a packet arriving from Host X1 to Router R1 must be forwarded to the next router on a path to X2.

2. **Routing.** The network layer must determine the route or path taken by data packets as they flow from a sender to a receiver. The algorithms that calculate these ride are referred to as routing algorithms. A routing algorithm would determine, for example, the path along which packets flow from X1 to X2.

Forwarding refers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface. Routing refers to the network-wide process that finds the end-to-end paths that packets take from source to destination.

Network Service Models:

The network service model defines the characteristics of end-to-end transport of packets between sending and receiving systems. In the sending host, when the transport layer passes a packet to the network layer, specific services that could be provided by the network layer include:

a. **Guaranteed delivery.** This service gives assurance that the packet will eventually arrive at its destination.

b. **Guaranteed delivery with bounded delay.** This service not only guarantees delivery of the packet, but delivery within a specified host-to-host delay bound.

The following services provided to a flow of packets between a source and destination:

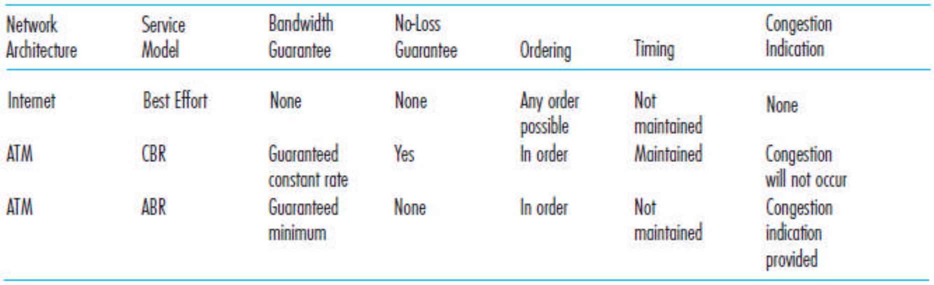
a. **In-order packet delivery.** This service guarantees that packets arrive at the destination in the order that they were sent.

b. **Guaranteed minimum bandwidth.** This network-layer service emulates the behavior of a transmission link of a specified bit rate between sending and receiving hosts. As long as the sending host transmits bits at a rate below the specified bit rate, then no packet is lost and each packet arrives within a pre specified host-to-host delay.

c. **Guaranteed maximum jitter.** This service guarantees that the amount of time between the transmission of two successive packets at the sender is equal to the amount of time between their receipt at the destination.

d. **Security services.** Using a secret session key known only by a source and destination host, the network layer in the source host could encrypt the payloads of all datagrams being sent to the destination host. In addition to confidentiality, the network layer could provide data integrity and source authentication services.

The Internet's network layer provides a single service, known as best-effort service. From Table 1, it might appear that best-effort service is a substitute for no service at all. With best-effort service, timing between packets is not guaranteed to be preserved, packets are not guaranteed to be received in the order in which they were sent, nor is the eventual delivery of transmitted packets guaranteed. Given this definition, a network that delivered no packets to the destination would satisfy the definition of best-effort delivery service.



Network Architecture	Service Model	Bandwidth Guarantee	No-Loss Guarantee	Ordering	Timing	Congestion Indication
Internet	Best Effort	None	None	Any order possible	Not maintained	None
ATM	CBR	Guaranteed constant rate	Yes	In order	Maintained	Congestion will not occur
ATM	ABR	Guaranteed minimum	None	In order	Not maintained	Congestion indication provided

Table 1. Internet, ATM CBR, and ATM ABR service models

Constant bit rate (CBR) ATM network service. This was the first ATM service model to be standardized, reflecting early interest by the telephone companies in ATM and the suitability of CBR service for carrying real-time, constant bit rate audio and video traffic. The goal of CBR service is conceptually simple to provide a flow of packets with a virtual pipe whose properties are the same as if a devoted fixed-bandwidth transmission link existed between sending and receiving hosts. With CBR service, a flow of ATM cells is carried across the network in such a way that a cell's end-to-end delay, the variability in a cell's end-to-end delay, and the selection of cells that are lost or delivered late are all guaranteed to be less than specified values. These values are acknowledged upon by the sending host and the ATM network when the CBR connection is first established.

Available bit rate (ABR) ATM network service. Internet offering so called best-effort service, ATM's ABR might best be characterized as being a slightly-better-than-best-effort service. As with the Internet service model, cells may be lost under ABR service. Cells cannot be reordered, and a minimum cell transmission rate (MCR) is guaranteed to a connection using ABR service. If the network has enough free resources at a given time, a sender may also be able to send cells successfully at a higher rate than the MCR.

3.3 NETWORK LAYER

3.3.1 Switching

Circuit Switching

Circuit switching, in which a physical circuit is established between the source and destination of the message before the delivery of the message. After the circuit is established, the entire message is transformed from the source to the destination. The source can then inform the network that the transmission is complete, which allows the network to open all switches and use the links and connecting devices for another connection.

In circuit switching, the whole message is sent from the source to the destination without being divided into packets.

Example: A good example of a circuit-switched network is the early telephone systems in which the path was established between a caller and a callee when the telephone number of the caller was dialed by the caller. When the callee responded to the call, the circuit was established. The voice message could now flow between the two parties, in both directions, while all of the connecting devices maintained the circuit. When the caller or callee hung up, the circuit was disconnected. The telephone network is not totally a circuit-switched network today.

Packet Switching

The network layer in the The Internet today is a packet-switched network. In this type of network, a message from the upper layer is divided into manageable packets and each packet is sent through the network. The source of the message sends the packets one by one; the destination of the message receives the packets one by one. The destination waits for all packets belonging to the same message to arrive before delivering the message to the upper layer. The connecting devices in a packet-switching network still need to decide how to route the packets to the final destination. Today, a packet-switched network can use two different approaches to route the packets: the datagram approach and the virtual circuit approach.

In packet switching, the message is first divided into manageable packets at the source before being transmitted. The packets are assembled at the destination

3.3.2 Packet Switching At Network Layer

The network layer is designed as a packet-switched network. This means that the packet at the source is divided into manageable packets, normally called datagrams. Individual datagrams are then transferred from the source to the destination. The received datagrams are assembled at the destination before recreating the original message. The packet-switched network layer of the Internet was originally designed as a connectionless service, but recently there is a tendency to change this to a connection oriented service. We first discuss the dominant trend and then briefly discuss the new one.

Connectionless Service

The network layer was designed to provide a connectionless service, in which the network layer protocol treats each packet independently, with each packet having no relationship to any other packet. The packets in a message may or may not travel the same path to their destination. When the Internet started, it was decided to make the network layer a connectionless service to make it simple. The idea was that the network layer is only responsible for delivery of packets from the source to the destination.

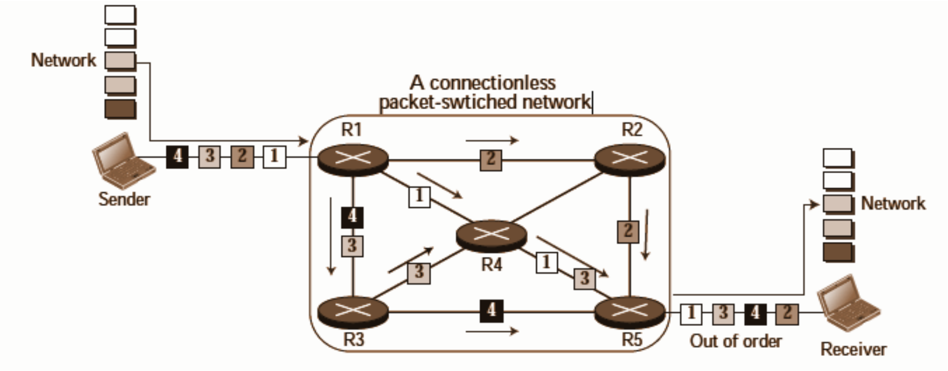


Fig. 2 A connectionless packet-switched network

Connection-Oriented Service

In a connection-oriented service, there is a connection between all packets belonging to a message. a virtual connection should be set up to define the path for the datagrams after that all datagrams in a message can be sent. After connection established, the datagrams can follow the same path. In this connection oriented service, the packet contains the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow. Figure 3 shows the concept of connection-oriented service.

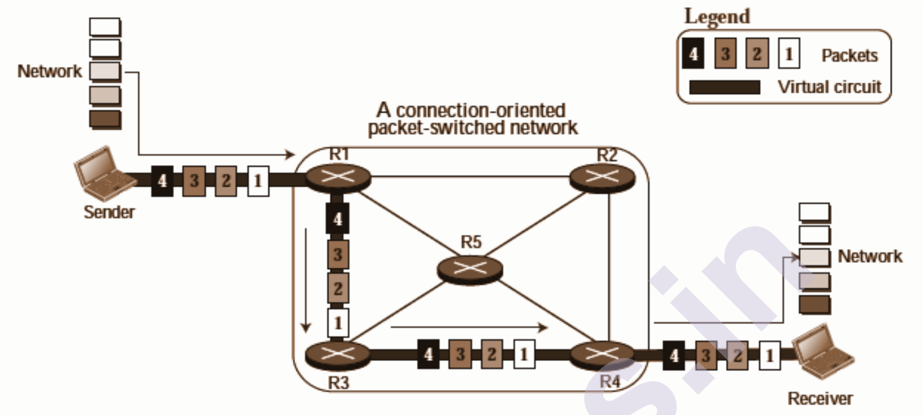


Fig. 3 connection-oriented packet switched network

3.4 VIRTUAL CIRCUIT AND DATAGRAM NETWORKS

Computer networks that provide only a connection service at the network layer are called virtual circuit networks, computer networks that provide only a connectionless service at the network layer are called datagram networks.

Characteristics:

1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
2. Resources can be assign during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.
3. As in a datagram network, data is packetized and each packet carries an address in the header. The reader may ask how the intermediate switches know where to send the packet if there is no final destination address carried by a packet.
4. Circuit-switched network, all packets follow the same path formed during the connection.
5. A virtual-circuit network is implemented in the data link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer.

Figure 4 is an example of a virtual-circuit network. The network has switches that allow traffic from sources to destinations. A source or destination can be a computer, packet switch, bridge, or any other device that connects other networks.

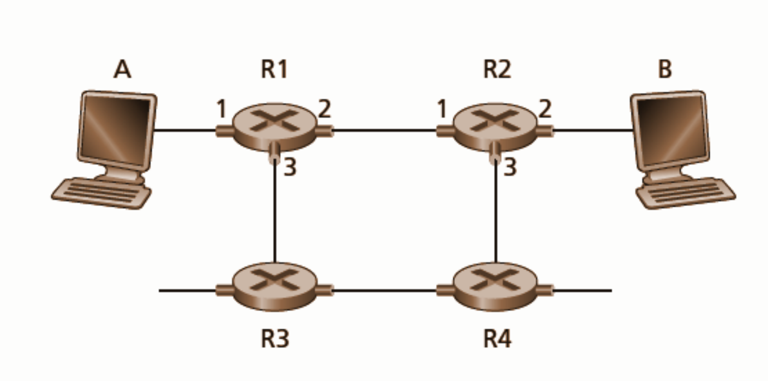


Fig. 4 A simple virtual circuit network

Virtual-Circuit Networks

ATM and frame relay are virtual-circuit networks and therefore, use connections at the network layer. These network-layer connections are called virtual circuits.

A virtual circuit consists of

1. a path between the source and destination hosts.
2. virtual circuit numbers, one number for each link as well the path, and
3. Updating the forwarding table in each router along the path.

A packet belonging to a virtual circuit will carry a virtual circuit number in its header. Because a virtual circuit may have a different virtual circuit number on each link, each intervening router must replace the virtual circuit number of each traversing packet with a new virtual circuit number. The new virtual circuit number is obtained from the forwarding table.

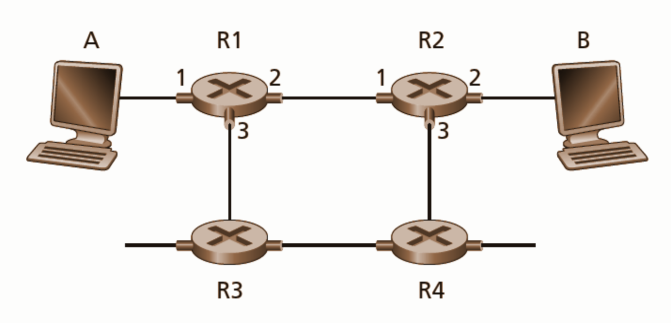


Fig. 5 A simple virtual circuit network

The numbers next to the links of R1 in Figure 5 are the link interface numbers. A requests that the network establish a virtual circuit between itself and Host B. Suppose also that the network chooses the path

A-R1-R2-B and assigns virtual circuit numbers 12, 22, and 32 to the three links in this path for this virtual circuit. In this case, when a packet in this virtual circuit leaves Host A, the value in the virtual circuit number field in the packet header is 12 when it leaves R1 Virtual Circuits, the value is 22; and when it leaves R2, the value is 32.

There are three identifiable phases in a virtual circuit:

1. **Virtual circuit setup.** during the setup phase, the sending transport layer contacts the network layer, specifies the receiver's address, and waits for the network to set up the virtual circuit . The network layer determines the path between sender and receiver, that is, the series of links and routers through which all packets of the virtual circuit will travel. The network layer also determines the virtual circuit number for each link along the path. At last, the network layer adds an entry in the forwarding table in each router along the path details.

2. **Data transfer.** As shown in Figure 6, once the virtual circuit has been established, packets can begin to flow along the virtual circuit .

3. **Virtual circuit teardown.** This is initiated when the sender informs the network layer of its desire to terminate the virtual circuit. The network layer will then typically inform the end system on the other side of the network of the call termination and update the forwarding tables in each of the packet routers on the path to indicate that the virtual circuit no longer exists.

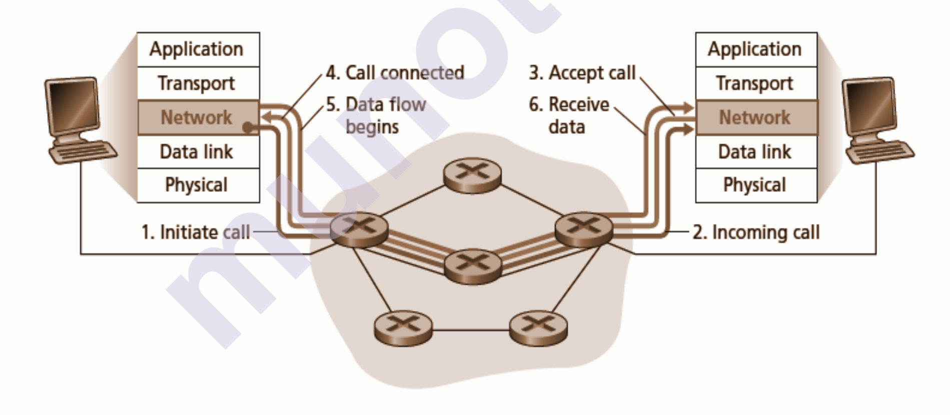


Fig. 6 Virtual-circuit setup

The messages that the end systems send into the network to initiate or terminate a Virtual Circuits, and the messages passed between the routers to set up the Virtual Circuits are known as signaling messages, and the protocols used to exchange these messages are often referred to as signaling protocols. Virtual Circuits setup is shown pictorially in Figure 6.

Datagram Networks:

In a datagram network, a packet is sent to an end system, it stamps the packet with the address of the destination end system and then pops the packet into the network. As shown in Figure 7, there is no Virtual Circuits setup and routers do not maintain any Virtual Circuits state information.

As a packet is transmitted from source system to destination system, it passes through a series of routers. All of these routers use the packet's destination address to forward the packet. especially.. Each router has a forwarding table that maps destination addresses to link interfaces. when a packet arrives at the router, the router uses the packet's destination address to look up the appropriate output link interface in the forwarding table. The router then purposely forwards the packet to that output link interface.

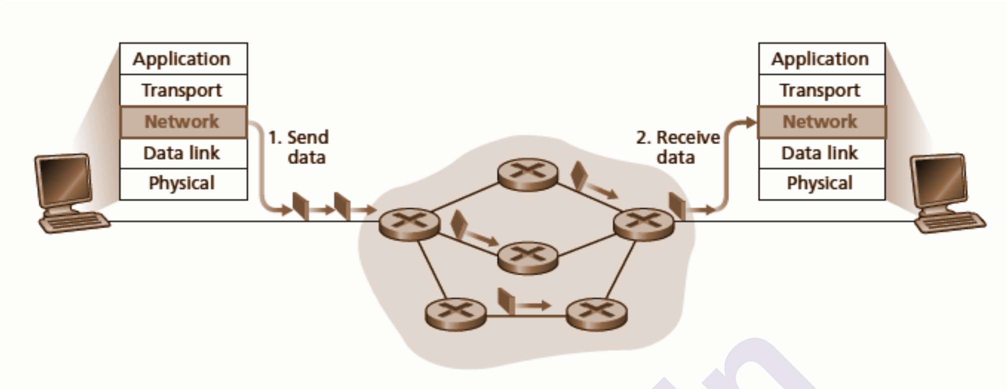


Fig. 7 Datagram network

3.5 NEED OF ROUTER

A router is a three layer device that routes packets according to their logical addresses. A router usually connects LANs and WANs on the Internet and has a routing table that is used for making decisions about the route. The routing tables are dynamic and are updated using routing protocols. Figure 1 shows a part of the Internet that uses routers to connect LANs and WANs.

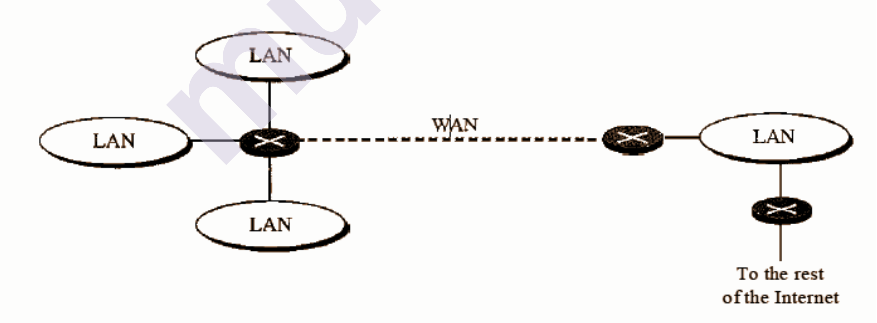


Fig. 1 Routers connecting independent LANs and WANs

Forwarding function, the actual transfer of packets from a router's incoming links to the appropriate outgoing links at that router. The terms forwarding and switching are often used interchangeably by computer-networking researchers and practitioners

A high-level view of a generic router architecture is shown in Figure 2 Four router components can be identified:

- **Input ports**
  An input port performs several key functions. It executes the physical layer function of terminating an incoming physical link at a router; this is shown in the leftmost box of the input port and the rightmost box of the output port in Figure.

- **Switching fabric**
  The switching fabric connects the router's input ports with its output ports. This switching fabric is totally contained within the router a network inside of a network router.

- **Output ports**
  An output port stores packets received from the switching fabric and sends these packets on the outgoing link by performing the necessary link-layer and physical-layer functions.

- **Routing processor**
  The routing processor accomplish the routing protocols, maintains routing tables and attached link state information, and computes the forwarding table for the router. It also performs the network management functions.

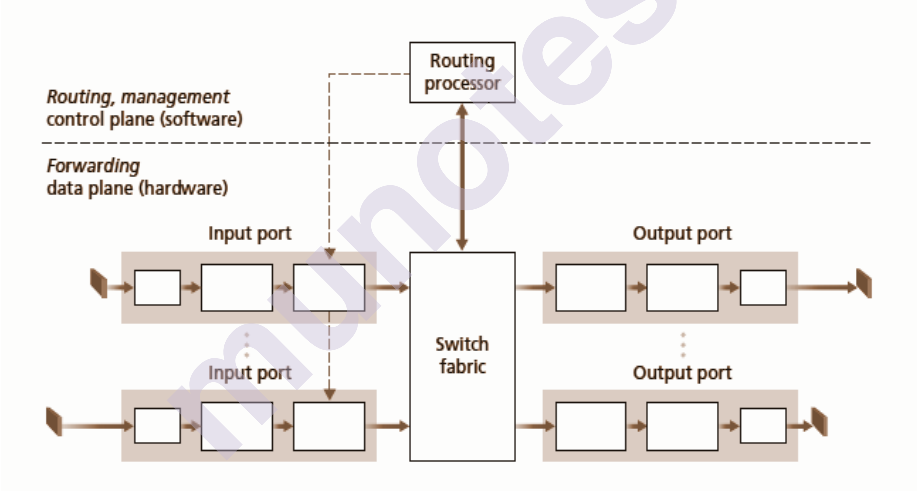


Fig. 2 Router architecture

3.6 THE INTERNET PROTOCOL (IP)

The function of IP, how addressing and forwarding are done on the Internet. The important components of the Internet Protocol (IP) is internet addressing and forwarding. There are two versions of IP. IP protocol version 4, which is usually referred to simply as IPv4. IP version 6, which has been proposed to replace IPv4.

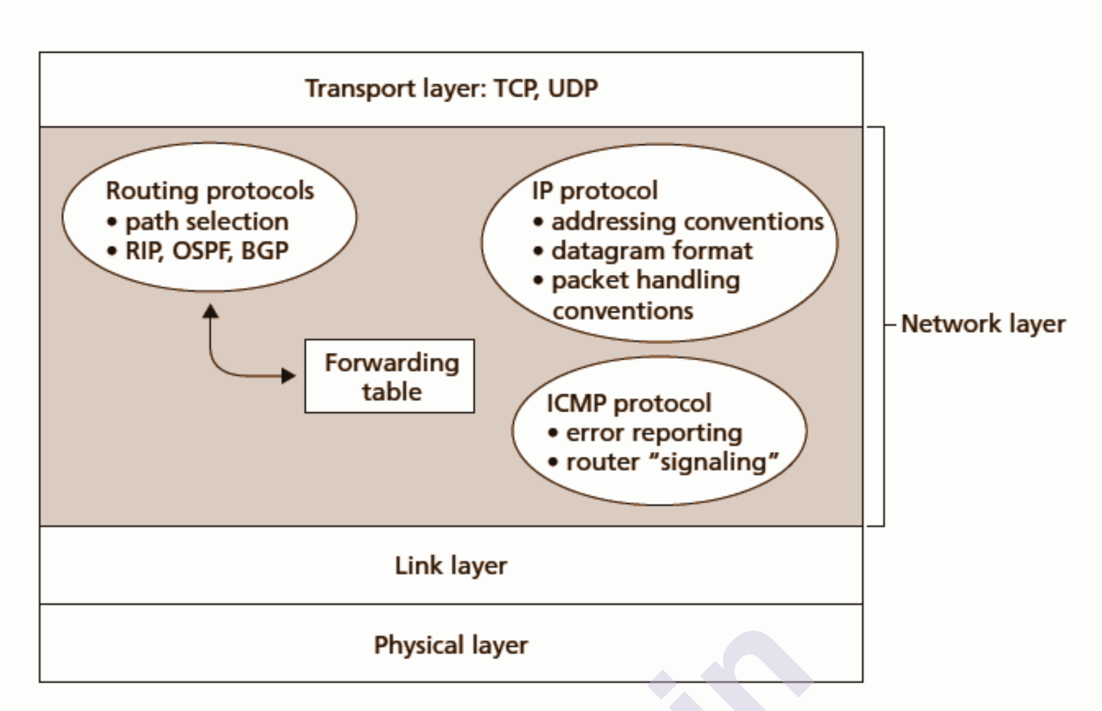


Fig. 3 A look inside the Internet's network layer

.

As shown in Figure 3, the Internet's network layer has three major components.

IP protocol: The first component is the IP protocol.

Routing Component: The second major component is the routing component, which determines the path a datagram follows from source to destination. routing protocols calculate the forwarding tables that are used to forward packets through the network.

ICMP Protocol: The final component of the network layer is a facility to report errors in datagrams and respond to requests for certain network-layer information.

3.6.1 Datagram Format

Network-layer packet is referred to as a datagram. The datagram plays a central role in the IPv4 datagram format is shown in Figure 4. The fields in the IPv4 datagram are the following:

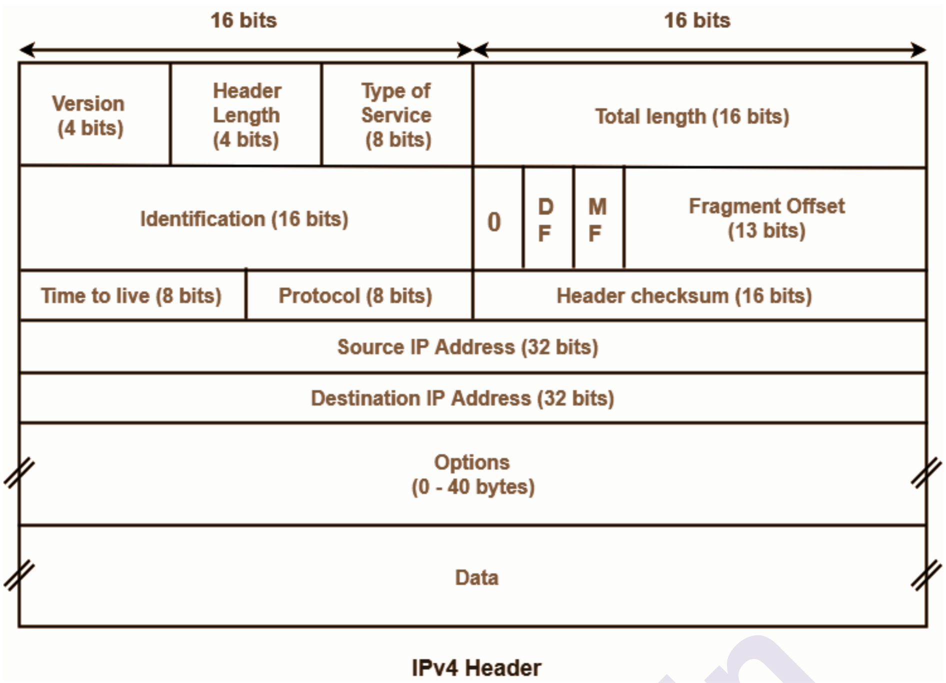


Fig. 4 IPv4 datagram format

- **Version number.** These 4 bits mentioned the IP protocol version of the datagram. The version number, the router can determine how to interpret the remainder of the IP datagram. Different versions of IP use different datagram formats.
- **Header length**. An IPv4 datagram can contain a variable number of options, 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 included in the IPv4 header to allow different types of IP datagrams to be distinguished from each other. For example, it might be useful to distinguish real-time datagrams from non-real-time traffic. The certain 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, measured in units of bytes. After all this field 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.** The Identification field (16 bits) is populated with an ID number unique for the combination of source & destination addresses and Protocol. the first, reserved bit of the Flags field (3 bits) will be 0 and the second bit, Don't Fragment, The **F**ragment Offset field (13 bits) is used to indicate the begin position of the data in the fragment in relation to the start of the data in the original packet.
- **Time-to-live**. When the 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 reduced by one each time the datagram is processed by a router. When the TTL field reaches 0, the datagram must be dropped.

- **Protocol.** This field is used only when an IP datagram reaches its end destination. The value of this field indicates the specific transport-layer protocol. For example, a value of 6 indicates that the data portion is passed to TCP, a value of 17 indicates that the data is passed to UDP.
- **Header checksum**. The header checksum supports a router in detecting bit errors in a received IP datagram. The header checksum is calculated by treating each 2 bytes in the header as a number and summing these numbers using 1s complement arithmetic.
- **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.
- **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. However, the minimal existence of options does complicate matters since datagram headers can be of variable length, one cannot determine a priori where the data field will start.
- **Data (payload)**. 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.

IP Datagram Fragmentation

All link-layer protocols can carry network-layer packets of the same size. different protocols can carry big datagrams, whereas other protocols can carry only little packets. For example, Ethernet frames can carry up to 1,500 bytes of data, whereas frames for some wide-area links can carry no more than 576 bytes. The maximum amount of data that a link-layer frame can transfer is called the maximum transmission unit (MTU).

The solution is to fragment the data in the IP datagram into two or more smaller IP datagrams, encapsulate each of these smaller IP datagrams in a separate link-layer frame and send these frames over the outgoing link. These smaller datagrams are referred to as a fragment.

Figure 5 an example. A datagram of 4,000 bytes data, 20 bytes of IP header and 3,980 bytes of IP payload. when it comes at a router and must be forwarded to a link with an MTU of 1,500 bytes of data. This understood that the 3,980 data bytes in the original datagram must be allocated to three separate fragments

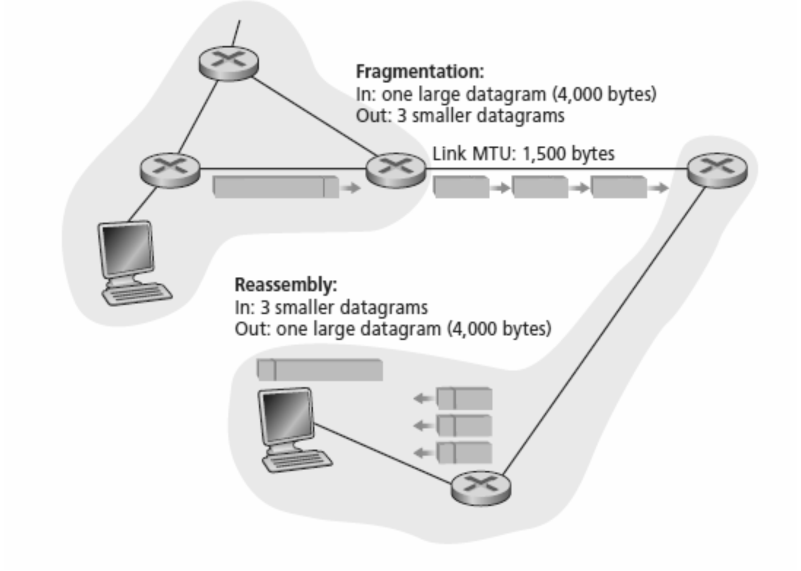


Fig. 5 IP fragmentation and reassembly

3.6.2 IPv4 Addressing

A host has only a single link into the network, when the IP in the host wants to send a datagram, it does so over this link. The wall between the host and the physical link is called an interface. Now consider a router and its interfaces. The function of a router is to receive a datagram on one link and forward the datagram on some other link, a router necessarily has two or more links to which it is connected to each other. The partition between the router and any one of its links is also called an interface. A router has multiple interfaces, one for each of its links. Every host and router is capable of sending and receiving IP datagrams, IP requires each host and router interface to have its own IP address. An IP address is technically connected with an interface, rather than with the host or router containing that interface. Each IP address is 32 bits long (4 bytes), and there are thus a total of 232 possible IP addresses. By approximating 210 by 103, it is easy to see that there are about 4 billion possible IP addresses. These addresses are written in dotted-decimal notation, in which each byte of the address is written in its decimal form and is separated by a period (dot) from other bytes in the address. For example, consider the IP address 193.32.216.9. The 193 is the decimal point equivalent of the first 8 bits of the address; the 32 is the decimal equivalent of the second 8 bits of the address, and so on. Thus, the address 193.32.216.9 in binary notation is

11000001 00100000 11011000 00001001

Figure 6 provides an example of IP addressing and interfaces. Here one router (with three interfaces) is used to interconnect seven hosts. The three hosts in the upper-left portion of Figure 6, and the router interface to which they are connected, all have an IP address of the form 223.1.1.xxx. This means that they all have the same leftmost 24 bits in their IP address.

The four interfaces are interconnected to each other by a network that contains no routers at all.

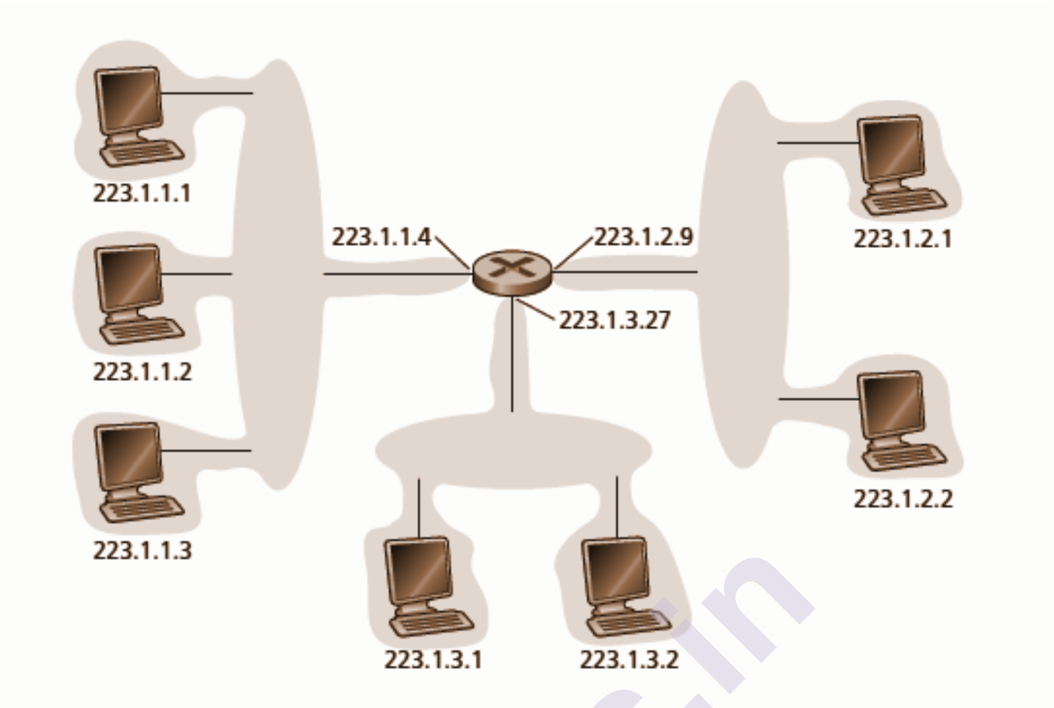


Fig 6. Interface addresses and subnets

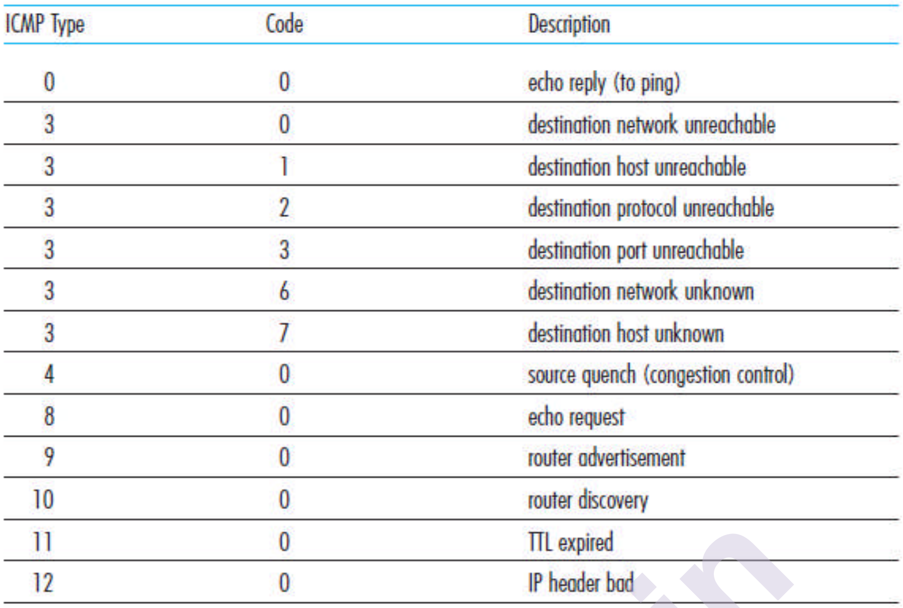
3.6.3 Internet Control Message Protocol (ICMP)

ICMP is used by hosts and routers to transfer network layer information to each other. The use of ICMP is for error reporting. For example, when running a Telnet, FTP, or HTTP session, you may have encountered an error message such as “Destination network unreachable.” This message had its origins in ICMP. At some point, an IP router was unable to find a path to the host specified in your Telnet, FTP, or HTTP application. That router generated and sent a type-3 ICMP message to your host indicating the error. ICMP often examines part of IP but architecturally it lies just above IP, as ICMP messages are carried inside IP datagrams.

That is, ICMP messages are carried as IP payload, just as TCP or UDP segments are carried as IP payload. Similarly, when a host receives an IP datagram with ICMP specified as the upper-layer protocol, it demultiplexes the datagram’s contents to ICMP, just as it would demultiplex the datagram's content to TCP or UDP. ICMP messages consist of a type and a code field, and contain the header and the first 8 bytes of the IP datagram that caused the ICMP message to be generated in the first place.

ICMP message types are shown in Figure 7 Note that ICMP messages are used not only for signaling error conditions. The ping program sends an ICMP type 8 code 0 message to the specified host. The destination host, glimpse the echo request, sends back a type 0 code 0

ICMP echo reply. Most TCP/IP implementations support the ping server directly in the operating system; that is, the server is not a process.



ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Fig. 7 ICMP message types

3.6.4 IPv6

Internet Protocol version 6 (IPv6) is the most recent version of thehttps://en.wikipedia.org/wiki/Internet_ProtocolInternet Protocol (IP), thehttps://en.wikipedia.org/wiki/Communication_protocolcommunications protocol that provides an identification and location system for computers on networks and routes traffic across the https://en.wikipedia.org/wiki/InternetInternet.

IPv6 Datagram Format

The format of the IPv6 datagram is shown in Figure 6. The most important updates introduced in IPv6 are evident in the datagram format:

- **Expanded addressing capabilities.**
  IPv6 increases the size of the IP address from 32 to 128 bits. In addition to unicast and multicast addresses, IPv6 has introduced a new type of address, called an anycast address, which allows a datagram to be delivered to any one of a group of hosts.

- **A streamlined 40-byte header.**
  A number of IPv4 fields have been released or made optional. The resulting 40 byte fixed length header allows for faster processing of the IP datagram message.

- **Flow labeling and priority.**
  labeling of packets belonging to particular flows for which the sender requests special handling, such as a non default quality of service or

real-time service. For example, audio and video transmission might likely be treated as a flow. On the other hand, the more traditional applications, such as file transfer and e-mail, might not be treated as flows.

The following fields are defined in IPv6:

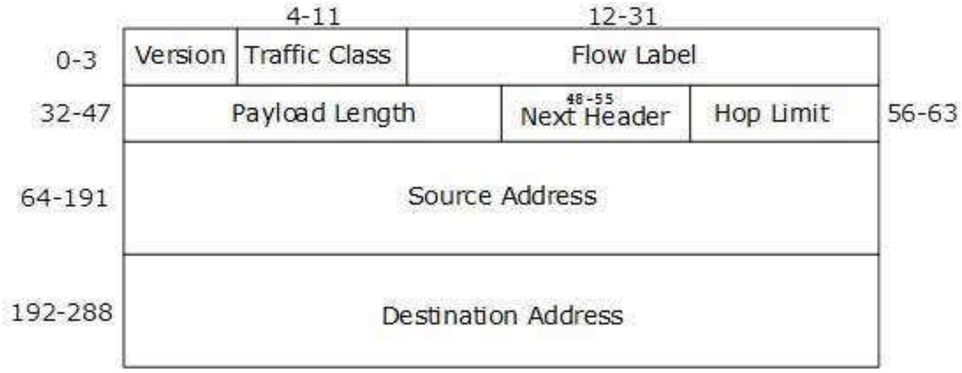


Fig 8. IPv6 header format

1. Version. This 4-bit field identifies the IP version number. IPv6 carries a value of 6 in this field. Note that putting a 4 in this field does not create a valid IPv4 datagram.
2. Traffic class. This 8-bit field is similar in spirit to the TOS field we saw in IPv4.
3. Flow label. this 20-bit field is used to identify a flow of datagrams.
4. Payload length. This 16-bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.
5. Next header. This field identifies the protocol to which the contents (data field) of this datagram will be delivered (for example, to TCP or UDP). The field uses the same values as the protocol field in the IPv4 header.
6. Hop limit. The contents of this field are decremented by one by each router that forwards the datagram. If the hop limit count reaches zero, the datagram is discarded.
7. Source and destination addresses. The various formats of the IPv6 128-bit address are described in RFC 4291
8. Data. This is the payload portion of the IPv6 datagram. When the datagram reaches its destination, the payload will be removed from the IP datagram and passed on to the protocol specified in the next header field.

3.7 SUMMARY

1. learned that a router may need to process millions of flows of packets between different source-destination pairs at the same time.
2. To permit a router to process such a large number of flows, network designers have learned over the years that the router's tasks should be as simple as possible.
3. A datagram network layer rather than a virtual-circuit network layer, using a streamlined and fixed-sized header, eliminating fragmentation and providing the one and only best-effort service.

3.8 REFERENCE FOR FURTHER READING

1. TCP/IP Protocol Suite 4 edition, Behrouz Forouzan, McGraw-Hill Science (2009)
2. Computer Networking: A Top-Down Approach 6th edition, James F. Kurose, Keith W. Ross, Pearson (2012).

3.9 UNIT END EXERCISES

1. Explain the services provided by Network Service Models?

2. Explain the difference between circuit switching and packet switching?

3. Write a short note on Virtual Circuit and Datagram Networks

❖❖❖❖

4

NETWORKING - IV

Unit Structure

4.1 OBJECTIVE

1. To help learners get an important distinction between the forwarding and routing functions of the network layer.
2. To understand the packet forwarding, a router at its hardware architecture and organization.
3. To learn the job of a routing algorithm is to determine good paths from senders to receivers.
4. To study the theory of routing algorithms.
5. To understand the broadcast and multicast routing.

4.2 INTRODUCTION

Routing is the process of selecting a pathway for traffic in a network or between or across multiple networks. In General, routing is performed in many types of networks, including circuit-switched networks, such as the public switched telephone network (PSTN), and computer networks, such as the Internet.

In packet switching networks, routing is the higher-level decision making that handles network packets from their source toward their destination through intermediate network nodes by specific packet forwarding mechanisms. Packet forwarding is the movement of network packets from one network interface to another. In-between nodes are typically network hardware devices such as routers, gateways, firewalls, or switches. Common-purpose computers also forward packets and perform routing, they have no specially optimized hardware for the task.

4.3 ROUTING ALGORITHMS & ROUTING IN THE INTERNET

In Today internet world, an internet can be so large that one routing protocol cannot manage the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems. An autonomous system (AS) is a group of networks and routers under the authority of a single administration.

Intra-domain routing:

Routing inside an autonomous system is referred to as intra-domain routing.

Inter-domain routing

Routing between two or more 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. Anyhow, only one interdomain routing protocol handles routing between autonomous systems.

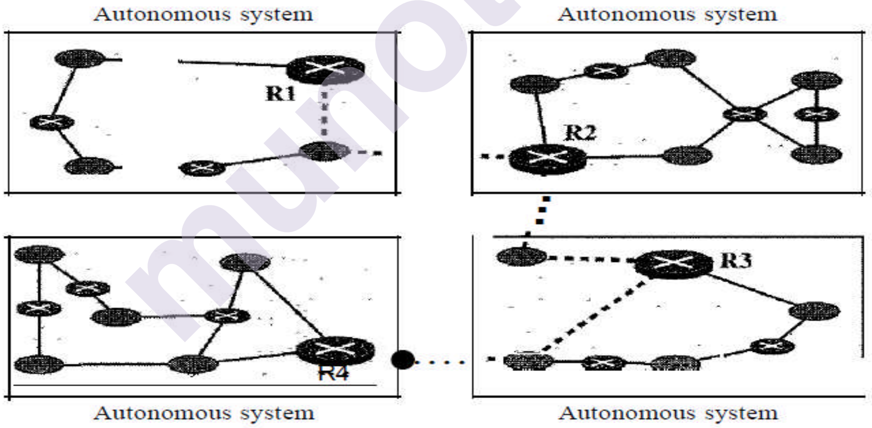


Fig. 9 Intra and inter domain routing

Intra-domain routing protocols:

1. Distance vector and
2. Link state.

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

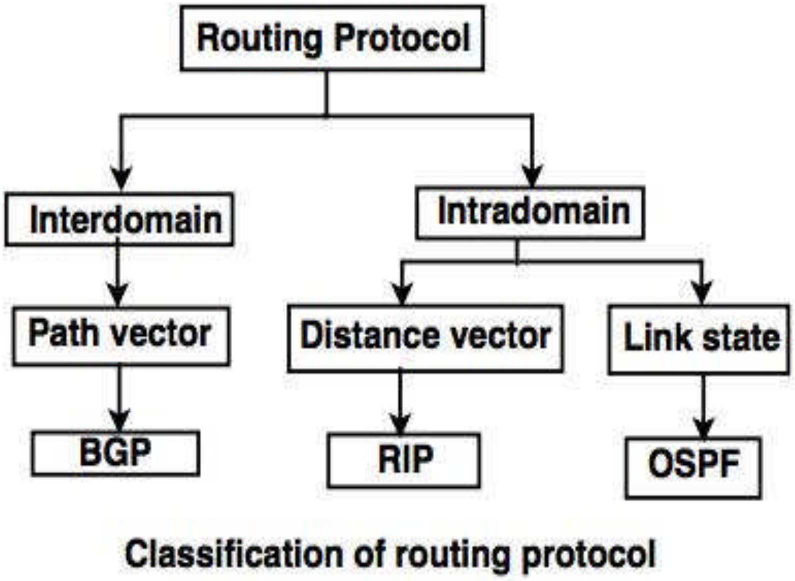


Fig. 10 Classification of routing protocol

Classify routing algorithms is according to whether they are global or decentralized.

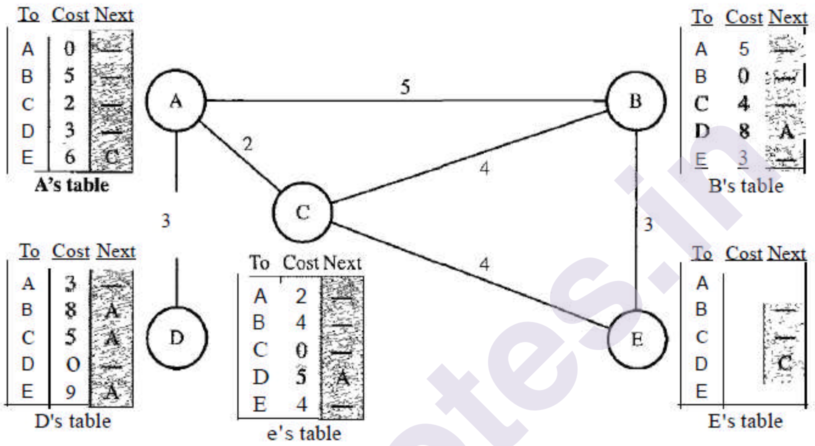
- A global routing algorithm calculates the least-cost path between a source and destination using complete, global knowledge about the network. This algorithm takes the connectivity between all nodes and all link costs as inputs parameter. This algorithm require somehow obtain this information before actually performing the calculation. The calculation itself can be run at one site or replicated at multiple sites. The feature is that a global algorithm has complete information about connectivity and link costs. In exercise, algorithms with global state information are frequently referred to as link-state algorithms, after all the algorithm must be aware of the cost of each link in the network.

- In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, share out manner. No one node has complete information about the costs of all network links. Rather than, each node begins with only the knowledge of the costs of its own directly attached links. Then, through an iterative process of calculation and interchange of information with its neighboring nodes, a node gradually calculates the least-cost path to a destination or set of destinations.

- A second broad way to classify routing algorithms is according to whether they are static or dynamic. Static routing algorithms, routes change very slowly over time, often as a result of human mediation dynamic routing algorithms change the routing paths as the network traffic loads or change of topology. A dynamic algorithm can be run either betimes or in direct response to topology or link cost changes. Dynamic algorithms are more responsive to network changes; they are also more susceptible to problems such as routing loops and oscillation in routes.

4.3.1 Distance Vector Routing

In distance vector routing, each node shares its routing table with its instant neighbors periodically and when there is a change. In distance vector routing, the least cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector table of minimum distances to every node. The table which at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).Example, nodes as the cities in an area and the lines as the roads connecting them. A vector table can show a tourist the minimum distance between cities. In Figure 11, we show a system of five nodes with their corresponding tables.



A's table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

B's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	8	A
E	3	—

D's table

To	Cost	Next
A	3	—
B	8	A
C	5	A
D	O	—
E	9	A

e's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

E's table

To	Cost	Next
A		
B		—
C		—
D		C
E		

Fig. 11 Distance vector routing tables

1. Initialization of DV table

This state is stable, each node knows how to reach any other node and the cost of that node. At the beginning, anyhow this is not the case. Each node can know only the distance between itself and its immediate neighbors, those directly connected to it through proper link. Let us assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors' nodes. Figure 11 shows the initial vector tables for each node. The distance for any entry that is not a neighbor is marked as an infinite loop.

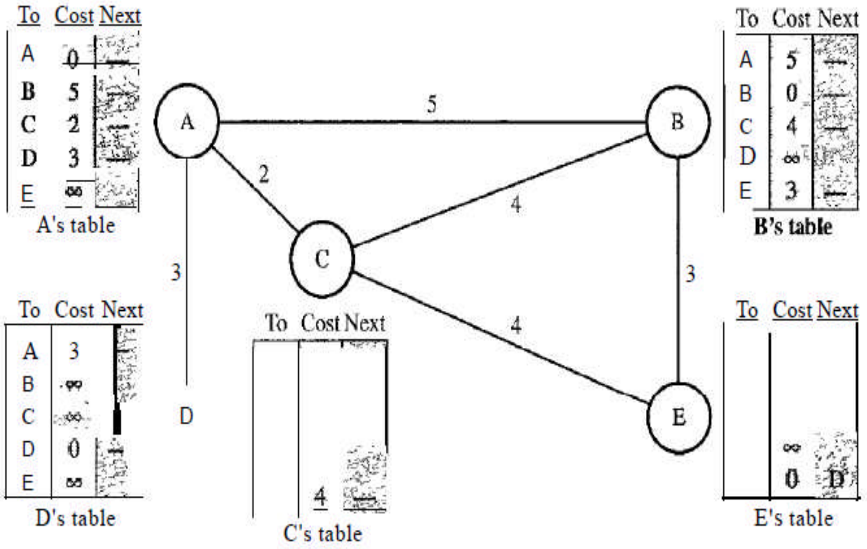


Fig. 12 Initialization of tables in distance vector routing

2. Sharing the vector table

The distance vector routing is the sharing of information (vector table) between neighbors nodes. Even through node A does not know about node E, node C does. So if node C shares its routing table with node A, node A can also know how to reach node E. On the other side, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbor's node, can improve their routing tables if they help each other. A node is not aware of a neighbor's node table. The best solution for each node is to send its entire table to the neighbor node and let the neighbor node decide what part to use and what part to discard.The third column of a routing table (next stop) is not useful for the neighbor node. When the neighbor node receives a table, this column needs to be replaced with the sender's name. If any of the rows can be used, the next node is the sender of the routing table. A node therefore can send only the first two columns of its table to any neighbor node.

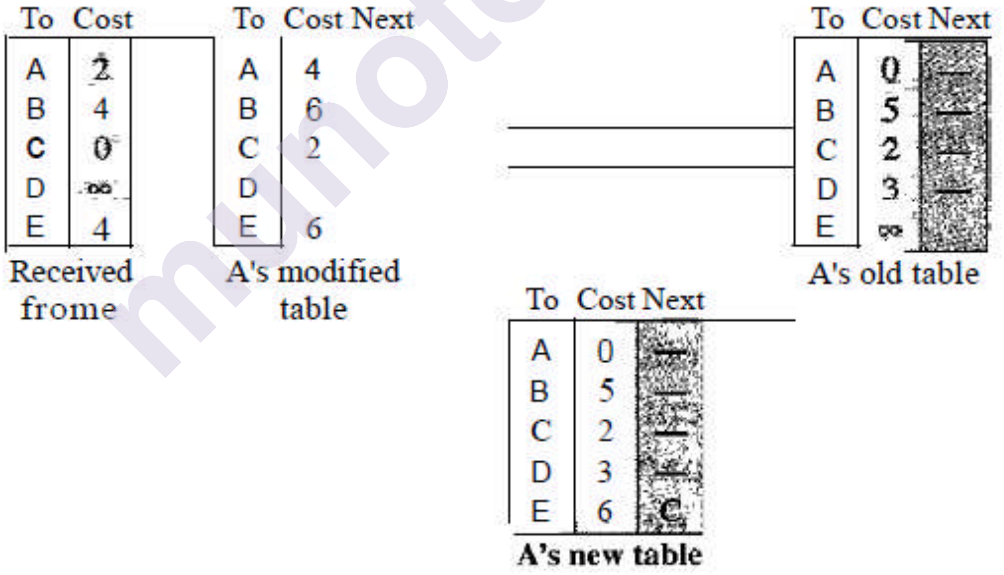
3. Updating the vector table

When a node receives a two-column table from a neighbor node, it needs to update its routing table. Updating of routing tables takes three steps:

1. The receiving node or routing table needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C hold that its distance to a destination is x mi, and the distance between A and C is y mi, then the distance between A and that destination, via C, is x + y mi.

2. The receiving node requires to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending/dispatch node is the next node in the route.
3. The receiving node requires comparing each row of its old table with the corresponding row of the modified version of the received table.
   a. If the next-node entry is different from one, the receiving node chooses the row with the smaller cost. If there is a bind, the old one is kept.
   b. If the next node appearance is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance between.
   c. For example that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller than the new one. The old route does not exist anymore. The new route has a distance of infinity.

Figure 13 shows how node A updates its routing table after receiving the partial table from node C.



To	Cost
A	2
B	4
C	0
D	∞
E	4

Received frome

To	Cost	Next
A	4	
B	6	
C	2	
D		
E	6	

A's modified table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's old table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A’s new table

Fig. 13 Updating in distance vector routing

Periodic Update A node table sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing algorithm. Triggered Update A node sends its two-column routing table to its neighbor’s node anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.

1. A node receives a table from a neighbor, resulting in changes in its own table after updating.
2. A node detects some failure in the neighboring links which results in a distance change to infinity.

RIP

The Routing Information Protocol (RIP) is an intradomain routing protocol used inside an autonomous system. It is a very straightforward protocol based on distance vector routing. RIP implements distance vector routing directly with some considerations:

1. In an autonomous system, dealing with routers and networks or links. The routers have routing tables, networks do not.
2. The destination in a routing table is a network, which means the first column defines a network address.
3. The metric used by RIP is very simple, the distance is defined as the number of links (networks) to reach the destination. For this reason, the metric in RIP protocol is called a hop count.
4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
5. The next-node tables column defines the address of the router to which the packet is to be sent to reach its destination.

4.3.2 Link State Routing

Link state routing has a different ideology from that of distance vector routing. In link state routing, if each node in the domain has the whole topology of the domain the list of nodes and links, they are connected each other including the type, cost (metric), and condition of the links (up or down) the node can use Dijkstra's algorithm to build a routing table. Figure 14 shows the concept.

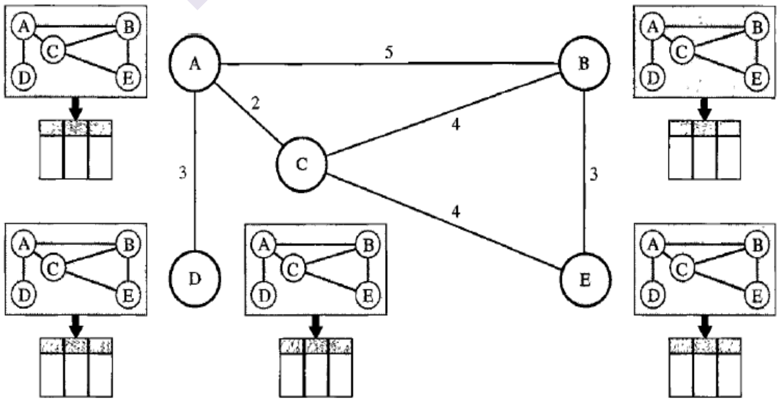


Fig. 14 Concept of link state routing

The above figure shows a simple domain with five nodes. Each node uses the same topology to create a routing table, but the routing table for each node is distinctive because the calculations are based on different interpretations of the topology. This is comparable to a city map. While each person may have the same map, each needs to take a different route to reach her specified destination.

Link state routing is based, the global knowledge about the topology is not clearly mentioned, and each node has partial knowledge about the same: it knows the state (type, condition, and cost) of its links. the whole topology can be compiled from the partial knowledge of each node. Figure 15 shows the same domain as in Figure 14, indicating the part of the knowledge belonging to each node.

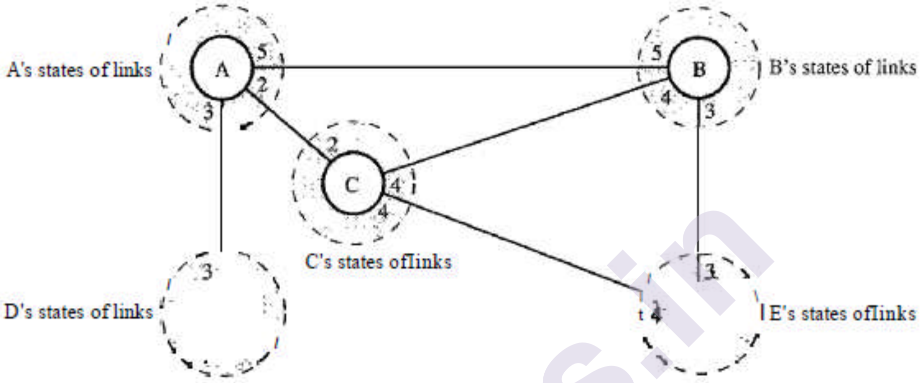


Fig. 15 Link state knowledge

Node A knows that it is connected to node B with metric 5, to node C with metric 2, and to node D with metric 3. Node C knows that it is connected to node A with metric 2, to node B with metric 4, and to node E with metric 4. Node D realizes that it is connected only to node A with metric 3. And so on. There is an overlap in the knowledge, the overlap assurance the creation of a common topology-a picture of the whole domain for each node.

Building Routing Tables

In link state routing, four sets of steps are required to confirm that each node has the routing table showing the least-cost node to every other node.

1. Design of the states of the links by each node, called the link state packet (LSP).
2. Circulation of LSPs to every other router, called flooding, in an efficient and reliable way.
3. Origin of a shortest path tree for each node.
4. Computation of a routing table based on the shortest path tree.

OSPF

The Open Shortest Path First OSPF protocol is an intradomain routing protocol based on the concept of link state routing. Its domain is also called an autonomous system. OSPF protocol divides an autonomous system into areas and subsections. An area is a group of networks, hosts, and routers all contained within an autonomous system. An autonomous system can be split into many different areas. All networks inside an area must be connected to each other through a link. Routers inside an area flood the area with the help of routing information. At the border of an area, special routers called area border routers. the areas inside an autonomous system is a special area called the backbone AS. All the areas inside an autonomous system must be connected to the backbone. The area identification of the backbone is zero. Figure 16 shows an autonomous system and its areas.

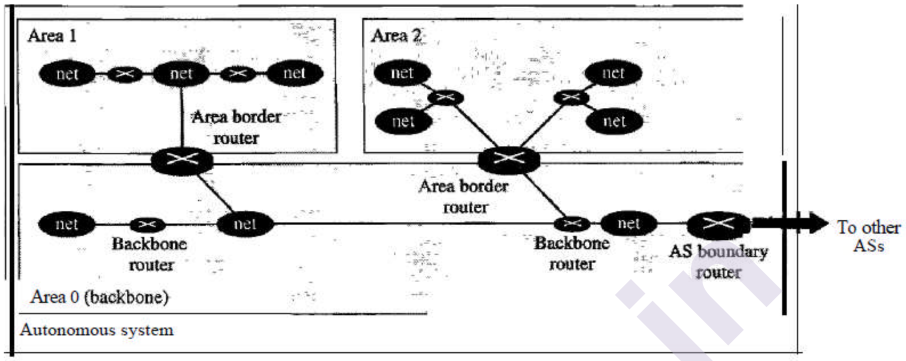
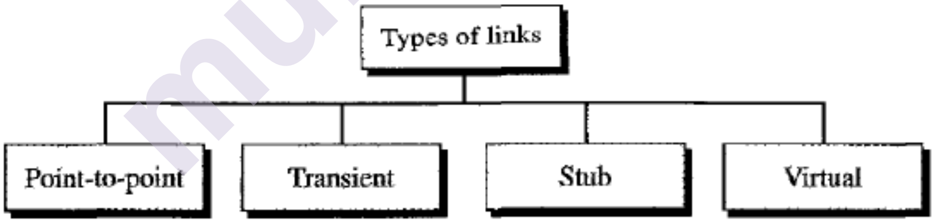


Fig. 16 Areas in an autonomous system

Types of Links in OSPF:

A connection is called a link. Four types of links have been defined:

a. point-to-point,
b. transient,
c. stub, and
d. virtual



a. A point-to-point link connects two routers without any help of any other host or router in between. An example of this type of link is two routers connected to each other by a telephone line or a T line. There is no need to assign a network address to this type of link.

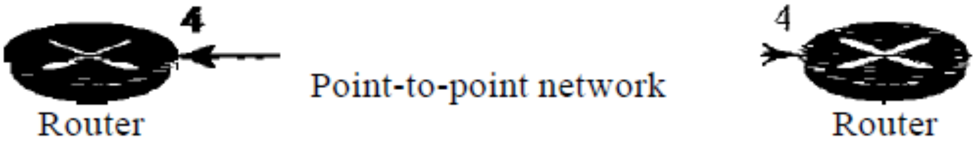


Fig. 17 Point-to-point link

b. A transient link is a network link with several routers attached with each other. The data can come into the network through any of the routers and leave through any router. For example, consider the Ethernet in Figure 18. Router A has routers B, C, D, and E as neighbor's node. Router B has routers A, C, D, and E as neighbor's nodes.

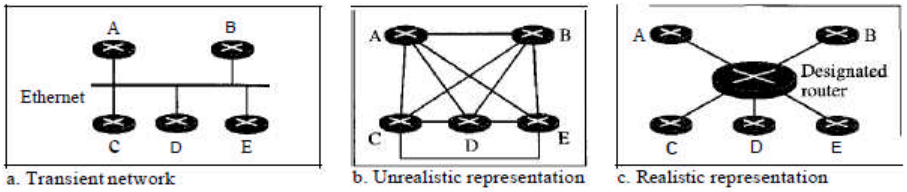


Fig. 18 Transient link

c. A stub link is a network that is connected to only one router in the network. The data packets come into the network through this single router and leave the network through this same router.

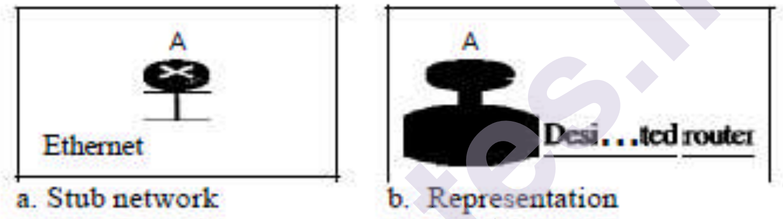


Fig.19 Stub link

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

4.3.3 Path Vector Routing

Distance vector routing is subject to uncertainty if there are more than a few hops in the domain of operation. Link state routing needs a large amount of resources to calculate routing tables. It also creates heavy traffic due to flooding. There is a need for a third routing protocol which we call path vector routing. Path vector routing demonstrates to be useful for interdomain routing. The idea of path vector routing is similar to that of distance vector routing.

BGP

Border Gateway Protocol is an interdomain routing protocol, It uses path vector routing. For example, a large business 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 called an autonomous system. This divides 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.

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.

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

External and Internal BGP

BGP has two types of sessions: external BGP (E-BGP) and internal BGP (I-BGP) sessions. The E-BGP session is used to interchange information between two speaker nodes. It belongs to two different AS. The I-BGP session is used to exchange routing information between two routers inside an AS. Figure 21 shows the details

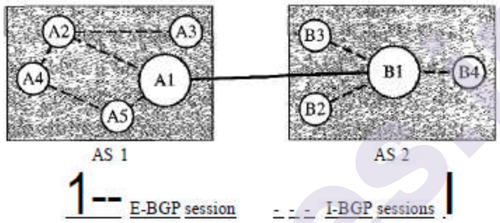


Fig. 21 Internal and external BGP sessions

BGP squint advertises routes to each other on the network. Two of the more important attributes are AS-PATH and NEXT-HOP:

AS-PATH:

This attribute contains the ASs through which the advertisement for the prefix has passed. When a prefix is passed into an AS, the AS adds its ASN to the ASPATH attribute.

NEXT-HOP:

Providing the critical link between the inter-AS and intra-AS routing protocols, the NEXT-HOP attribute has a subtle but important use. The NEXT-HOP is the router interface that begins the AS-PATH.

BGP Route Selection

In AS BGP uses eBGP and iBGP to distribute routes to all the routers. From this issue, a router may learn about more than one route to any one prefix, in which case the router must select one of the possibleroutes. The input into route selection process is the set of all routes that have been acquire and accepted by the router. If there are more than two routes to the same prefix, then BGP sequentially invokes the following elimination rules until one route remains:

1. Routes are assigned a local priority value as one of their attributes. The local priority of a route could have been set by the router. This is a policy decision that is boost up to the AS's network administrator. The routes with the giant local preference values are selected.

2. The route with the shortest path (AS-PATH) is selected. If this rule applies the only rule for route selection, then BGP would be using a DV algorithm for path determination, where the distance metric uses the number of AS hops preferably than the number of router hops.

3. From the remaining routes the route with the closest NEXT-HOP router is selected. Here, closest means cost of the least-cost path, set on by the intra-AS algorithm, is the smallest

4. If more than one route still leftover, the router uses BGP identifiers to select the route.

4.4 SUMMARY

1. A router may need to process millions of flows of packets between different source-destination pairs at the same time.
2. We learned how routing algorithms abstract the computer network to a graph with nodes and links.
3. We learn that there are two broad approaches: a centralized approach, in which each node obtains a complete map of the network and independently applies a shortest-path routing algorithm; and a decentralized approach, in which individual nodes have only a partial picture of the entire network, yet the nodes work together to deliver packets along the shortest routes.
4. We learned how centralized, decentralized, and hierarchical approaches are embodied in the principal routing protocols in the Internet: RIP, OSPF, and BGP.

4.5 REFERENCE FOR FURTHER READING

1. TCP/IP Protocol Suite 4 edition, BehrouzForouzan, McGraw-Hill Science (2009)
2. Computer Networking: A Top-Down Approach 6th edition, James F. Kurose,Keith W. Ross, Pearson (2012).

4.6 UNIT END EXERCISES

1. Why are different inter-AS and intra-AS protocols used in the Internet?
2. Why are policy considerations as important for intra-AS protocols, such as OSPF and RIP, as they are for an inter-AS routing protocol like BGP?
3. How does BGP use the NEXT-HOP attribute? How does it use the AS-PATH attribute?
4. What is an important difference between implementing the broadcast abstraction via multiple unicasts, and a single network- (router-) supported broadcast?

❖❖❖❖

Module - II

5

NETWORK VIRTUALIZATION

Unit Structure

5.0 OBJECTIVES:

This is an introductory tutorial, which covers the basics of Virtualization and explains how to deal with its various components and sub-components.

Introduction:

Virtualization is utilization of computer resources which is not in used 100%.

Virtualization is technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system. Software called a hypervisor connects directly to that hardware and allows you to split 1 system into separate, distinct, and secure environments known as virtual machines (VMs).

Virtualization is technology that lets you create useful IT services using resources that are traditionally bound to hardware. It allows you to use a physical machine's full capacity by distributing its capabilities among many users or environments.

5.2 AN OVERVIEW:

Virtualization is a technology that helps users to install different Operating Systems on a hardware. They are completely separated and independent from each other.

5.3 NEED FOR VIRTUALIZATION

Sometimes it's necessary to make a virtual machine of any operating system like (Linux or windows) into an existing and running base operating system on a standalone hardware. Creation of VMs may differ as per need and requirements.

The most important function of virtualization is the capability of running multiple operating systems and applications on a single computer or server. This means increased productivity achieved by fewer servers.

Following figure (i) illustrate the need of the virtualization.

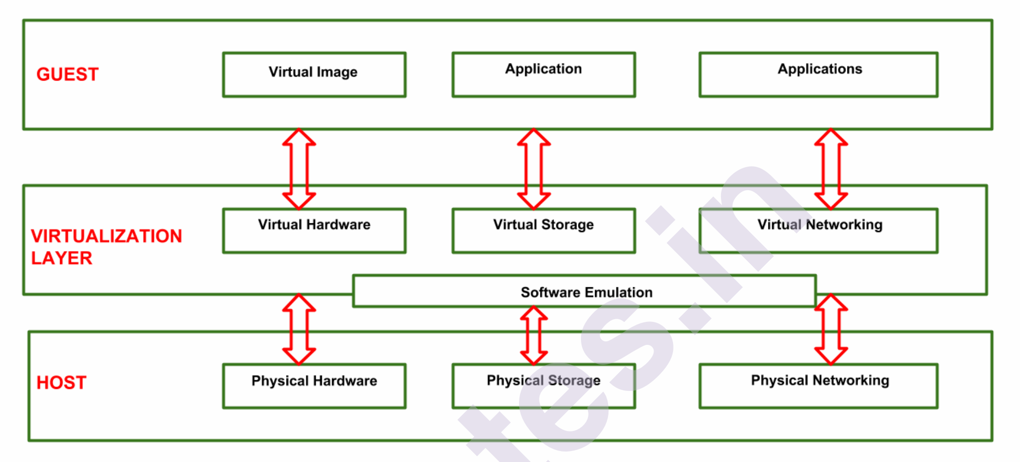


Fig (1-1)

The virtual Enterprise

"A **virtual enterprise is** a temporary alliance of **enterprises** that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation **is** supported by computer networks. "

Many business functions of your organization can be outsourced. What traditionally were considered core functions are no longer a sacred territory and are available for outsourcing. The difference in cost and efficiency between an "on demand" or pay per usage outsourced service and an on-premises and self-manned typical function could be significant and hard to ignore.

This presents a problem requiring a solution for an Enterprise that outsources most or all its business functions but retains governance for planning, coordinating operations, budgeting, and making all key decisions. In a Wikipedia definition, "a virtual organization is a firm that outsources the majority of its functions." The Virtual Enterprise (VE) can be successful, assuming it employs best of breed outsourced services in a "virtual" Value Chain implementation consisting of company and partner links.

A VE operates over a virtual Value Chain, i.e., a chain whose links are owned by a company and its partners, blurring the borders between the Value Chain of the firm and the Value Network it is a part of.

The Governance is the business function that defines and identifies the Virtual Enterprise, since most or all other functions of the Enterprise (primary and secondary in Porter's definition) could be outsourced.

The VE is defined by a new operating model promoting collaboration and B2B to take advantage of best of breed applications on the market. This VE business model is increasingly achievable by the adoption of business process outsourcing (BPO), application outsourcing – Software as a Service (SaaS) – and, in general, by the fast adoption of infrastructure virtualization technologies, Web Services, SOA, and collaborative technologies of the Web2.0.

The "Virtual" Enterprise could be the darling of the entrepreneurial world, specializing in management and governance skills while outsourcing most of the Functions of the Enterprise today.

Transport Virtualization-VNs

The authors of Network Virtualization define the technical requirements posed by the need to virtualize the network. Based on these requirements, they propose an architectural framework comprised of the functional areas necessary to successfully support concurrent virtual networks (VNs) over a shared enterprise physical network.

When segmenting the network pervasively, all the scalability, resiliency, and security functionality present in a non-segmented network must be preserved and in many cases improved. As the number of groups sharing a network increases, the network devices must handle a much higher number of routes. Any technologies used to achieve virtualization must therefore provide the necessary mechanisms to preserve resiliency, enhance scalability, and improve security.

Central Services Access: Virtual Network Perimeter

The default state of a VN is to be totally isolated from other VNs. In this respect, VNs could be seen as physically separate networks. However, because VNs actually belong to a common physical network, it is desirable for these VNs to share certain services such as Internet access, management stations, DHCP services, *Domain Name System* (DNS) services, or server farms. These services will usually be located outside of the different VNs or in a VN of their own. So, it is necessary for these VNs to have a gateway to connect to the "outside world." The outside world is basically any network outside the VN such as the Internet or other VNs. Because this is the perimeter of the VN, it is also desirable for this perimeter to be protected by security devices such as firewalls and *intrusion detection systems* (IDSs). Typically, the perimeter is deployed at a common physical location for most VNs. Hence, this location is known as the central services site, and the security devices here deployed can be shared by many VNs.

The creation of VNs could be seen as the creation of security zones, each of which has a unique and controlled entry/exit point at the VN perimeter. Routing within the VNs should be configured so that traffic is steered to the common services site as required. Figure illustrates a typical perimeter deployment for multiple VNs accessing common services. Because the services accessed through the VN perimeter are protected by firewalls, we refer to these as "protected services.

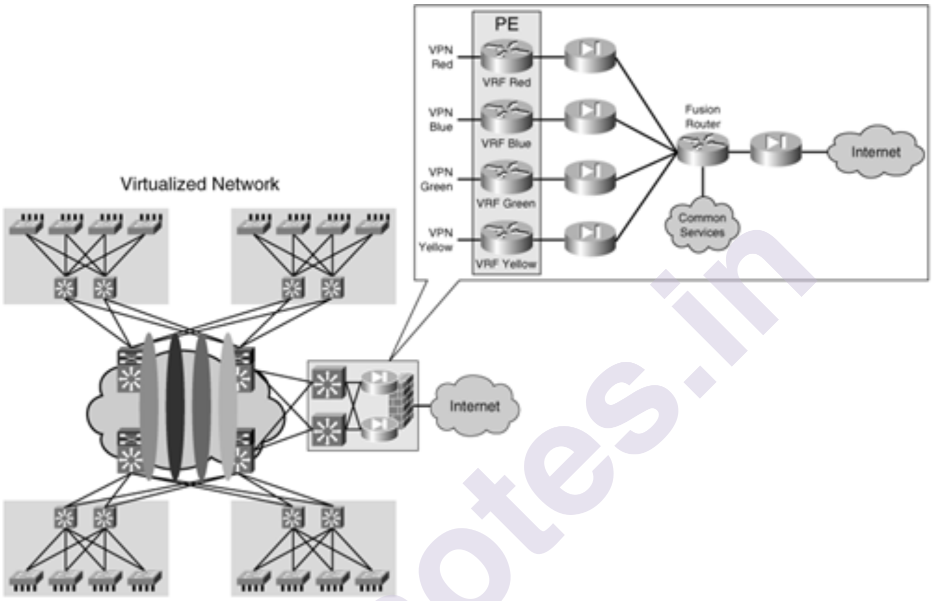


fig-(1-2)

As shown in above Figure, each VN is head ended by a dedicated firewall. This allows the creation of security policies specific to each VN and independent from each other. To access the shared services, all firewalls are connected to a "fusion" router. The fusion router can provide the VNs with connectivity to the common services, the Internet, or even inter-VN connectivity. The presence of this fusion router should raise two main concerns:

- The potential for traffic leaking between VNs
- The risk of routes from one VN being announced to another VN

The presence of dedicated per-VN firewalls prevents the leaking of traffic between VNs through the fusion router by only allowing established connections (connections initiated from "inside" the firewall) to return through the VN perimeter. It is key to configure the routing on the fusion device so that routes from one VN are not advertised to another through the fusion router. The details of the routing configuration at the central site are discussed in Chapter 8, "Traffic Steering and Service Centralization."

Figure shows an additional firewall separating the fusion area from the Internet. This firewall is optional. Whether to use it or not depends on the need to keep common services or transit traffic in the fusion area protected from the Internet.

A Virtualization Technologies primer: theory

- **Devices**—How is traffic separation maintained internally to a device? What are the primitives used for Layer 2, Layer 3, or Layer 4 traffic?
- **Data path**—How is traffic separation enforced across a network path? What tools are available to maintain the separation across a network?
- **Control plane**—Because data-path virtualization essentially builds an overlay topology, what changes are needed for routing protocols to function correctly?

5.4 NETWORK DEVICE VIRTUALIZATION

Network Virtualization (NV) refers to abstracting network resources that were traditionally delivered in hardware to software. NV can combine multiple physical networks to one virtual, software-based network, or it can divide one physical network into separate, independent virtual networks.

Network virtualization software allows network administrators to move virtual machines across different domains without reconfiguring the network. The software creates a network overlay that can run separate virtual network layers on top of the same physical network fabric.

One of the characteristics of a VN is that it provides what are essentially private communication paths between members of a group over a shared infrastructure. This creates two requirements for the network infrastructure:

- **Traffic from one group is never mixed with another**—For sending and receiving traffic over shared links, tunnels (many borrowed from existing *virtual private network* [VPN] solutions) can guarantee data separation. Network devices need to enforce group separation in their internal memory (for example, during routing table lookups, access lists processing or NetFlow statistics gathering).

- **Each VN has a separate address space**—This requirement is derived from the fact that VNs offer the same characteristics as a physical network. Address space and forwarding within it are two of the most basic aspects of any network.

Why Network Virtualization?

Network virtualization is rewriting the rules for the way services are delivered, from the software-defined data center (SDDC), to the cloud, to the edge. This approach moves networks from static, inflexible, and inefficient to dynamic, agile, and optimized. Modern networks must keep up with the demands for cloud-hosted, distributed apps, and the increasing threats of cybercriminals while delivering the speed and agility you need for faster time to market for your applications. With network virtualization, you can forget about spending days or weeks provisioning the infrastructure to support a new application. Apps can be deployed or updated in minutes for rapid time to value.

How does network virtualization work?

Network virtualization decouples network services from the underlying hardware and allows virtual provisioning of an entire network. It makes it possible to programmatically create, provision, and manage networks all in software, while continuing to leverage the underlying physical network as the packet-forwarding backplane. Physical network resources, such as switching, routing, firewalling, load balancing, virtual private networks (VPNs), and more, are pooled, delivered in software, and require only Internet Protocol (IP) packet forwarding from the underlying physical network.

Network and security services in software are distributed to a virtual layer (hypervisors, in the data center) and “attached” to individual workloads, such as your virtual machines (VMs) or containers, in accordance with networking and security policies defined for each connected application. When a workload is moved to another host, network services and security policies move with it. And when new workloads are created to scale an application, necessary policies are dynamically applied to these new workloads, providing greater policy consistency and network agility.

Benefits of network virtualization

Network virtualization helps organizations achieve major advances in speed, agility, and security by automating and simplifying many of the processes that go into running a data center network and managing networking and security in the cloud. Here are some of the key benefits of network virtualization:

- Reduce network provisioning time from weeks to minutes
- Achieve greater operational efficiency by automating manual processes
- Place and move workloads independently of physical topology
- Improve network security within the data center

Example:

One example of network virtualization is virtual LAN (VLAN). A VLAN is a subsection of a local area network (LAN) created with software that combines network devices into one group, regardless of physical location. VLANs can improve the speed and performance of busy networks and simplify changes or additions to the network.

Another example is network overlays. There are various overlay technologies. One industry-standard technology is called virtual extensible local area network (VXLAN). VXLAN provides a framework for overlaying virtualized layer 2 networks over layer 3 networks, defining both an encapsulation mechanism and a control plane. Another is generic network virtualization encapsulation (GENEVE), which takes the same concepts but makes them more extensible by being flexible to multiple control plane mechanisms.

VMware NSX Data Center – Network Virtualization Platform VMware NSX Data Center is a network virtualization platform that delivers networking and security components like firewalling, switching, and routing that are defined and consumed in software. NSX takes an architectural approach built on scale-out network virtualization that delivers consistent, pervasive connectivity and security for apps and data wherever they reside, independent of underlying physical infrastructure.
The first problem to solve is how to virtualize the forwarding plane in a way that meets the requirements for address and traffic flow separation. Depending on the type of device, the virtual separation can go by the following names:

- Virtual LAN (VLAN)
- Virtual routing and forwarding (VRF)
- Virtual forwarding instance (VFI)
- Virtual firewall context

Layer 2: VLANs

VLANs are a good example of a piece of the virtualization puzzle that has been around for quite some time. A VLAN is a logical grouping of ports on a switch that form a single broadcast domain. Ports in a VLAN can communicate only with other ports in the same VLAN. How a given switch does this is implementation dependent, but a common solution is for the switch to tag each frame with a VLAN number as it arrives on a port. When a frame is sent to other ports, the output hardware copies the packet only if it is configured with the VLAN number carried in the frame.

The summary effect of the VLANs is to partition the switch into logical Layer 2 domains. Each domain has its own address space and packets from one domain are kept separate from those of another.

Layer 3: VRF Instances

VRFs are to Layer 3 as VLANs are to Layer 2 and delimit the domain of an IP network within a router. The Cisco website has a more formal definition:

VRFA VPN Routing/Forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.

Unlike the VLAN scenario, where an extra column in the MAC table is adequate, a VRF partitions a router by creating multiple routing tables and multiple forwarding instances. Dedicated interfaces are bound to each VRF.

Following diagram shows a simple logical representation of a router with two VRFs: RED and GREEN. The RED table can forward packets between interfaces E1/0, E1/2, and S2/0.102. The GREEN table, on the other hand, forwards between interfaces E4/2, S2/0.103, and S2/1.103. An interface cannot be in multiple VRFs at the same time.

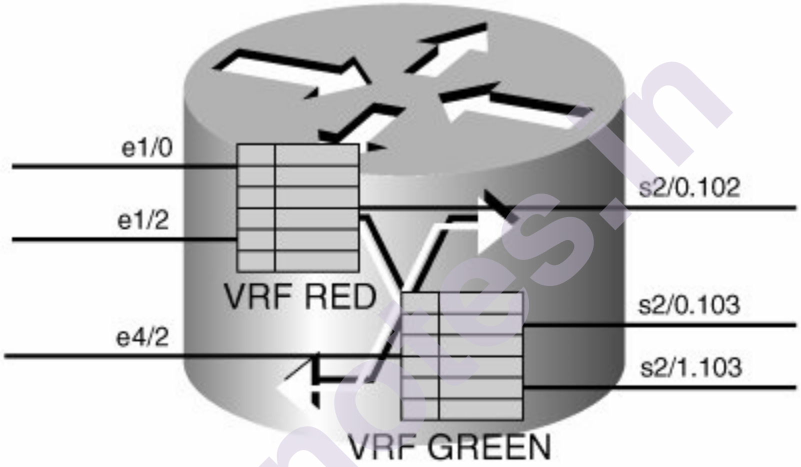


Fig (2-1). Multiple VRFs on a Router

FIBs and RIBs

Before looking at the routing information for a VRF, we need to introduce the routing table's two main data structures, which are used to find the egress interface for a given packet: theForwarding Information Base (FIB) and the Routing Information Base (RIB). Long gone are the days when a router maintained a single routing table on which it did linear, longest-prefix searches against destination IP addresses.

The FIB is a database of information used to forward packets. When a packet is received on a routed interface, the router looks up the destination address in the FIB to find the next hop for the packet.

The FIB structure is particularly efficient for resolving longest-prefix matches, and Cisco IOS resolves all route redirections so that a single lookup can yield the entry for the next hop of a packet. Cisco literature often mentions an adjacency concept when presenting the FIB. An adjacency is any node in the network that is reachable with a single Layer 2 hop. It so happens that Cisco IOS also maintains a data structure of adjacencies, which contains, among other things, interface and MAC layer rewrite information for all possible next hops. FIB entries point to the adjacency table, and in the remainder of this chapter, we group the two together and refer simply to the FIB. Hardware-based forwarding paths use the FIB concept, as does Cisco Express Forwarding (CEF).

Because it contains both Layer 2 and Layer 3 information, the FIB can be updated by several sources, such as routing protocol and Address Resolution Protocol (ARP) updates.

The RIB is the memory structure that contains classic routing data. The RIB can contain recursive routes. If a packet destination is not in the FIB, the router "punts" the packet to a slow processing path and resolves the destination next hop using the RIB.

Traffic processing happens according to the same rules as on a device with no VRFs:

1. Traffic enters the router.
2. The ingress policy is applied.
3. Routing and forwarding lookup occurs.
4. The egress policy is applied.
5. Traffic is forwarded.

Obviously, the ingress and egress policies can include QoS statements that prioritize traffic to or from a particular interface or address, but the fact that a packet belongs to a particular VRF has no impact on those policies. It simply alters what happens in Step 3 of the preceding list. It is far more common to want to bind an interface or packet flow to a particular VRF based on policy criteria. For example, all interfaces from a certain user domain are bound to a single company VRF, or packets with a 10.0.0.0/8 source address are bound to a guest VRF. We look at this in great detail in some of the design chapters.

Virtual and Logical Routers

A VRF is not the same thing as a completely virtualized device, even if they are sometimes confused as such (it is true that some marketing literature encourages such confusion). They simply allow routers to support multiple address spaces. This is some distance from a fully virtualized device, where resources can be more or less arbitrarily allocated to tasks.

Virtualized devices do exist, however, and, to cut through the fog of confusion, it is helpful to have a taxonomy of terms to start with:

- A logical router (LR) uses hardware partitioning to create multiple routing entities on a single device. An LR can run across different processors on different cards of a router. All the underlying hardware and software resources are dedicated to an LR. This includes network processors, interfaces, and routing and forwarding tables. LRs provide excellent fault isolation but do require abundant hardware to implement.

- A virtual router (VR) uses software emulation to create multiple routing entities. The underlying hardware is shared between different router processes (note that we mean an entire instance of something like the nonkernel parts of IOS, not a single router process). In a well-implemented virtual router, users can see and change only the configuration and statistics for "their" router.

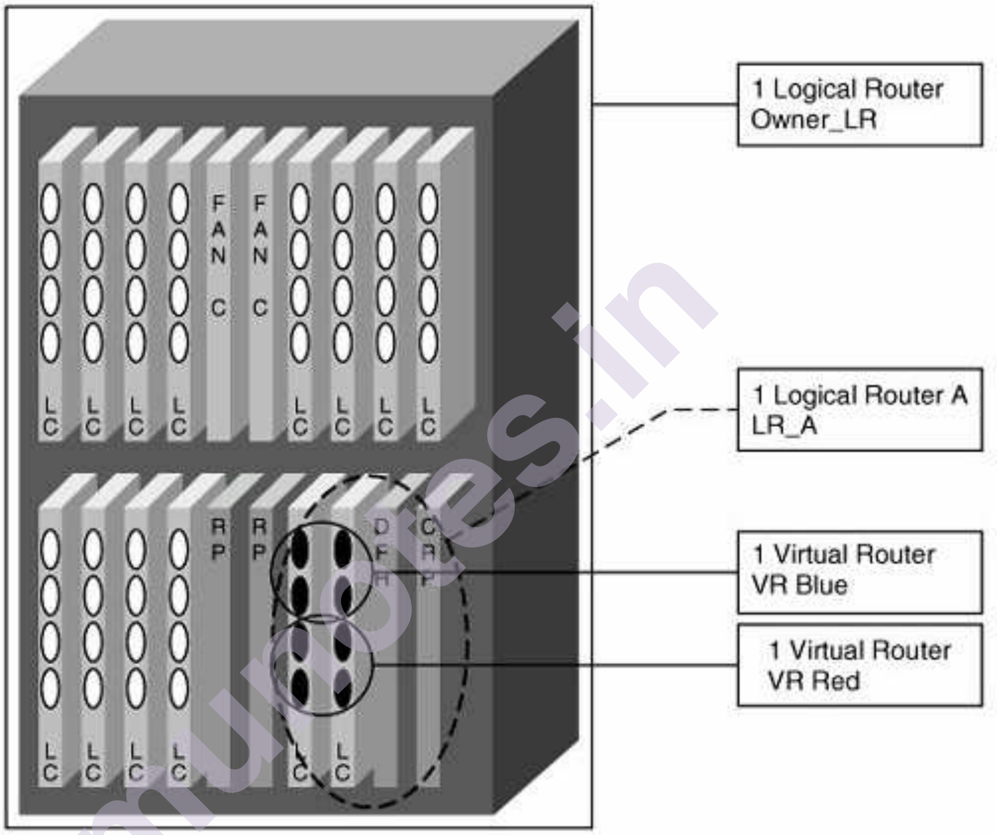


Fig (2-2) Logical and Virtual Routers

From the preceding list and mk:@MSITStore:F:\ \Idol\M.Sc-CS%20IDOL%20COURCE%20WRITING%20CONTENT%20FOR%20ANC\Network%20Virtualization%20by%20Victor%20Moreno,%20Kumar%20Reddy%20(z-lib.org).chm::/1587052482/ch04lev1sec1.html - ch04fig02, which gives a pictorial idea of the difference between VRs and LRs, you can see that only the LR is completely virtualized. Because of the cost involved of having all that extra hardware and device management, LRs tend to be high-end systems. A VR is a software-based virtualization solution, where all the tasks share the same hardware resources.

Layer 2 Again: VFIs

VFI is a service-specific partition on a switch that associates attachment circuits in the form of VLANs with virtual switched interfaces (VSIs).

If that did not make much sense, it is useful to have some background on the service itself, namely Virtual Private LAN Services (VPLS), to understand VFIs.

VPLS is a Layer 2 LAN service offered by service providers (SPs) to connect Ethernet devices over a WAN. The customer devices (call them customer edges [CEs] for now, are all Ethernet switches. However, the SP uses a Layer 3 network running Multiprotocol Label Switching (MPLS) to provide this service. The device on the edge of the SP network is called a provider edge (PE). Its role is to map Ethernet traffic from the customer LAN to MPLS tunnels that connect to all the other PEs that are part of the same service instance. The PEs are connected with a full mesh of tunnels and behave as a logical switch, called a VSI. Another way to think about this is to see the VPLS service as a collection of Ethernet ports connected across a WAN. A VSI is a set of ports that forms a single broadcast domain.

In many ways, a VSI behaves just as you would expect a regular switch to. When a PE receives an Ethernet frame from a customer device, it first learns the source address, as would any switch, before looking at the destination MAC address and forwarding the frame. If the port mapping for the destination MAC address is unknown, or is a broadcast, the frame is sent to all PEs that are part of the VSI. The PEs use split horizon to avoid creating loops, which in turn means that no spanning tree is needed across the SP network.

Obviously, the previous explanation hides a fair amount of detail, but it should be enough to give a high-level view of what is going on.

Once again, there is a need to define and manage groups of isolated ports and tunnels on a switch. The VLAN construct is too limited, and a VRF is strictly a Layer 3 affair, so it is necessary to come up with a new virtual device structure for VPLS, called a VFI.

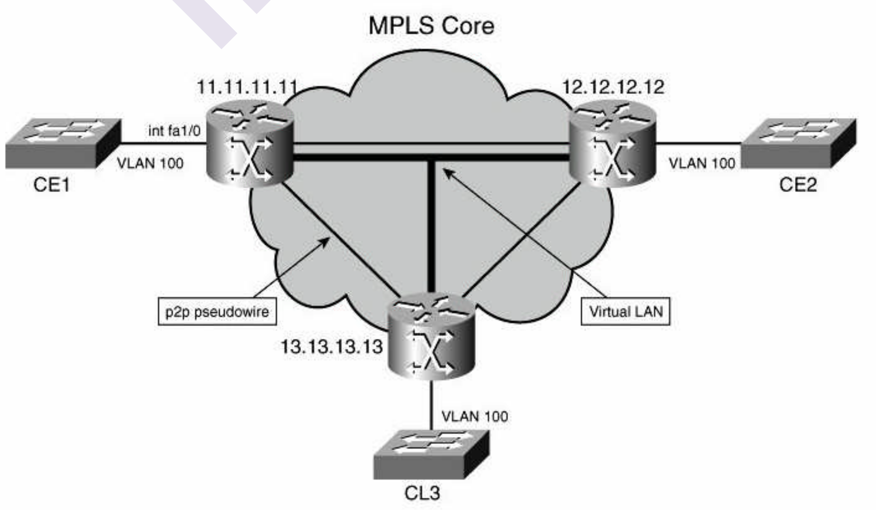


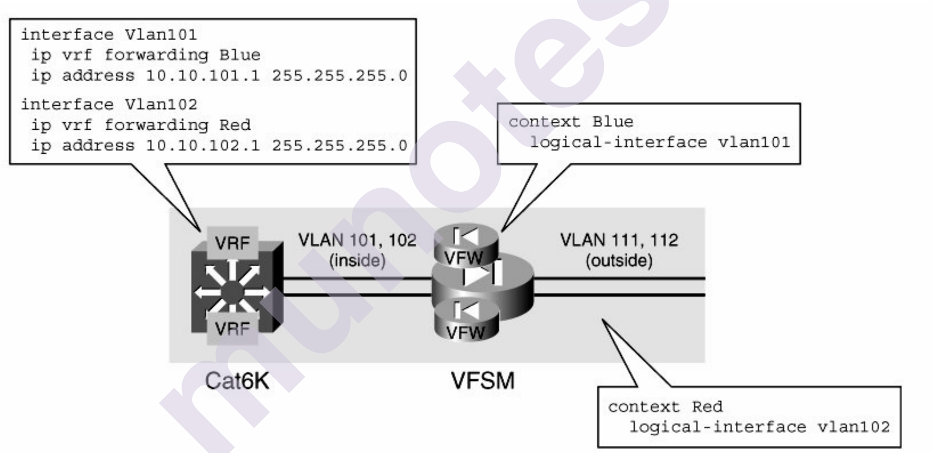
Fig (2-3).VPLS Topology

Virtual Firewall Contexts

Device virtualization is not limited to switches and routers. As a final example, consider a firewall device. For essentially economic reasons, you might want to share a single firewall between multiple different customers or network segments. Each logical firewall needs to have a complete set of policies, dedicated interfaces for incoming and outgoing traffic, and users authorized to manage the firewall.

Many vendors provide this capability today and undoubtedly have their own, well-chosen name for it, but on Cisco firewalls the term context is used to refer to a virtual firewall. Unlike VRFs, VFIs, or VLANs, a context is an emulation of a device.

Firewall contexts are a little unusual in the way they assign a packet to a context. All the partitions we have seen up to now have static assignment of interfaces (you can assign IP packets to a VRF dynamically. We cover that later). A firewall module looks at an incoming packet's destination IP address or Ethernet VLAN tag to decide which context a packet belongs to. All the firewall needs for one of the two fields to be unique. So, either each context has a unique IP address space on its interfaces or the address space is shared, but each context is in a different VLAN.



```
interface Vlan101
 ip vrf forwarding Blue
 ip address 10.10.101.1 255.255.255.0
interface Vlan102
 ip vrf forwarding Red
 ip address 10.10.102.1 255.255.255.0
```

```
context Blue
  logical-interface vlan101
```

```
context Red
  logical-interface vlan102
```

Fig (2-4).VRF on Switch Connected to Firewall Contexts Across VLANs

Network Device Virtualization Summary

True device virtualization allows resources to be allocated to tasks, or applications. We looked at four different primitives that virtualize the forwarding paths on switches or routers: VLAN and VFI for Layer 2, VRF for Layer 3, and contexts for firewalls. Each of these functions slightly differently. VRFs have the most extensive tie-ins with other features, which we use extensively in the design sections. Before covering data-path virtualization, one word about data center designs. We are focusing on network devices exclusively in this book and do not address the details of server and storage virtualization, which are two important topics in their own right.

Data-Path Virtualization

It refers to the virtualization of the interconnection between devices. This could be a single-hop or multiple-hop interconnection. For example, an Ethernet link between two switches provides a single-hop interconnection that can be virtualized by means of 802.1q VLAN tags; for Frame Relay or ATM transports, separate virtual circuits provide data-path virtualization. An example of a multiple-hop interconnection would be that provided by an IP cloud between two devices. This interconnection can be virtualized through the use of multiple tunnels (*generic routing encapsulation* [GRE] for example) between the two devices.

Layer 2: 802.1q Trunking

You probably do not think of 802.1q as a data-path virtualization protocol. But, the 802.1q protocol, which inserts a VLAN tag on Ethernet links, has the vital attribute of guaranteeing address space separation on network interfaces.

Obviously, this is a Layer 2 solution, and each hop must be configured separately to allow 802.1q connectivity across a network. Because a VLAN is synonymous with a broadcast domain, end-to-end VLANs are generally avoided.

Generic Routing Encapsulation

GRE provides a method of encapsulating arbitrary packets of one protocol type in packets of another type (the RFC uses the expression X over Y, which is an accurate portrayal of the problem being solved). The data from the top layer is referred to as the payload. The bottom layer is called the delivery protocol. GRE allows private network data to be transported across shared, possibly public infrastructure, usually using point-to-point tunnels.

Although GRE is a generic X over Y solution, it is mostly used to transport IP over IP (a lightly modified version was used in the Microsoft Point-to-Point Tunneling Protocol [PPTP] and, recently, we are seeing GRE used to transport MPLS). GRE is also used to transport legacy protocols, such as Internetwork Packet Exchange (IPX) and AppleTalk, over an IP network and Layer 2 frames.

GRE is purely an encapsulation mechanism. How packets arrive at tunnel endpoints is left entirely up to the user. There is no control protocol, no session state to maintain, no accounting records, and so forth; and this conciseness and simplicity allows GRE to be easily implemented in hardware on high-end systems. The concomitant disadvantage is that GRE endpoints have no knowledge of what is happening at the other end of the tunnel, or even whether it is reachable.

The time-honored mechanism for detecting tunnel reachability problems is to run a dynamic routing protocol across the tunnel. Routing Protocol (RP) keepalives are dropped if the tunnel is down, and the RP itself will declare the neighbor as unreachable and attempt to route around it. You can lose a lot of data waiting for an RP to detect a problem in this way and reconverge. Cisco added a keepalive option to its GRE implementation. This option sends a packet through the tunnel at a configurable period. After a certain number of missed keepalives (the number is configurable), the router declares the tunnel interface as down. A routing protocol would detect the interface down event and react accordingly.

GRE's lack of control protocol also means that there is essentially no cost to maintaining a quiescent tunnel active. The peers exchange no state information and must simply encapsulate packets as they arrive. Furthermore, like all the data-path virtualization mechanisms we discuss, the core network is oblivious of the number of tunnels traversing it. All the work is done on the edge.

We do not want to suggest that GRE is the VPN equivalent of a universal solvent. There is a cost to processing GR Eencapsulation / decapsulation, route lookup, and so forth but it's in the data path.

GRE IOS Configuration

On Cisco devices, GRE endpoints are regular interfaces. This seemingly innocuous statement is replete with meaning, because anything in Cisco IOS that needs to see an interface (routing protocols, access lists, and many more) will work automatically on a GRE tunnel.

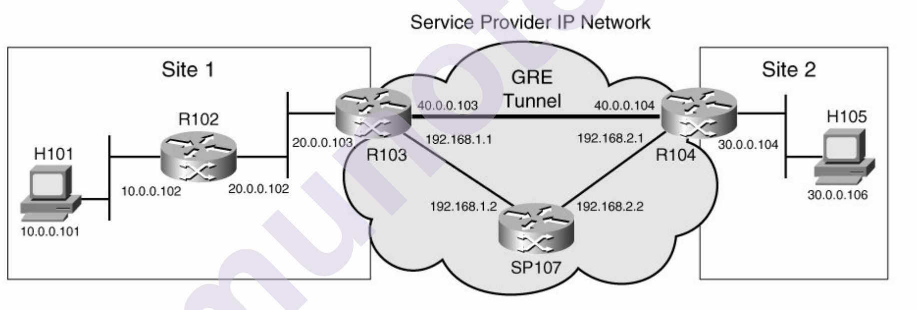


Fig (2-5).GRE Topology

The tunnel source and tunnel destination addresses are part of the transport network address space. They need to match on both endpoints so that a source address on one router is the destination address on the remote device. The router must also have a path in its routing table to the tunnel destination address. The next hop to the tunnel destination must point to a real interface and not the tunnel interface.

In this case, the router has a tunnel interface with tunnel destination of 192.168.2.1 on the public network. The 40.0.0.0/24 network used for the tunnel IP's address, however, is part of the private address space used on Sites 1 and 2.

IPsec

IPsec provides a comprehensive suite of security services for IP networks. IPsec was originally conceived to provide secure transport over IP networks. The security services include strong authentication (Authentication Header [AH]) and Encryption (Header [EH]) protocols and ciphers and key-exchange mechanisms. IPsec provides a way for peers to interoperate by negotiating capabilities and keys and security algorithms.

IPsec peers maintain a database of security associations. A security association (SA) is a contract between peers, which defines the following:

- The specific encryption and authentication algorithms used, such as Triple DES (Triple Data Encryption Standard)
- The IPsec protocol service (Encapsulating Security Payload [ESP] or AH)
- Key material needed to communicate with the peer

The SA is negotiated when an IPsec session is initiated. Each IPsec header contains a unique reference to the SA for this packet in a Security Parameter Index (SPI) field, which is 32-bit numeric reference to the SA needed to process the packet. Peers maintain a list of SAs for inbound and outbound processing. The value of the SPI is shared between peers. It is one of the things exchanged during IPsec session negotiation.

At the protocol level, there are two IPsec headers:

- AH Offers nonrepudiatable authentication between two parties. The authentication service also provides for message integrity and certain instances of (identity) spoofing.
- ESP Offers encrypted communication between two parties. The encryption service allows message confidentiality, integrity, nonrepudiation, and protection against spoofing and replay attacks.

It is possible to use authentication and encryption services separately or together. If used in combination, the AH header precedes the ESP header.

In a normal routing scenario, when a router needs to forward a packet, it finds the outgoing interface by looking for a matching IP address prefix in the routing table. The actual interface used for forwarding corresponds to the shortest path to the IP destination, as defined by the routing policy. Other administrative policies, such as QoS and security, may affect the choice of interface. This collection of criteria used for forwarding decisions is more generally referred to as a Forward Equivalency Class (FEC). The classification of a packet to FEC is done on each router along the IP path and happens independently of the other routers in the network.

MPLS decouples packet forwarding from the information in the IP header. An MPLS router forwards packets based on fixed-length labels instead of matching on a variable-length IP address prefix. The label is a sort of shortcut for an FEC classification that has already happened. Where the label comes from is discussed later in this section, but for now, it is enough to say that the labels are calculated based on the topology information in the IP routing table. RFC 3031 puts it like this:

In MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. The FEC to which the packet is assigned is encoded as a short fixed length value known as a "label." When a packet is forwarded to its next hop, the label is sent along with it; that is, the packets are "labeled" before they are forwarded.

In the MPLS forwarding paradigm, once a packet is assigned to a FEC, no further header analysis is done by subsequent routers; all forwarding is driven by the labels.

Before looking at this in more detail, we need to introduce some definitions:

- Label switching router (LSR) A router that switches based on labels. An LSR swaps labels. Unlike a traditional router, an LSR does not have to calculate where to forward a packet based on the IP packet header (which is a simplified way of saying it does not do FEC classification when it receives a packet). An LSR uses the incoming label to find the outgoing interface (and label). LSRs are also called provider (P) routers.
- Edge LSR A router that is on the edge of an MPLS network. The edge LSR adds and removes labels from packets. This process is more formally called imposition and disposition (and also pushing and popping, because labels are said to go on a stack). Edge LSRs are often referred to as provider edge (PE) routers.

Customer edge (CE) An IP router that connects to the PE device. The CE performs IP forwarding. The PE and CE form routing protocol adjacencies.

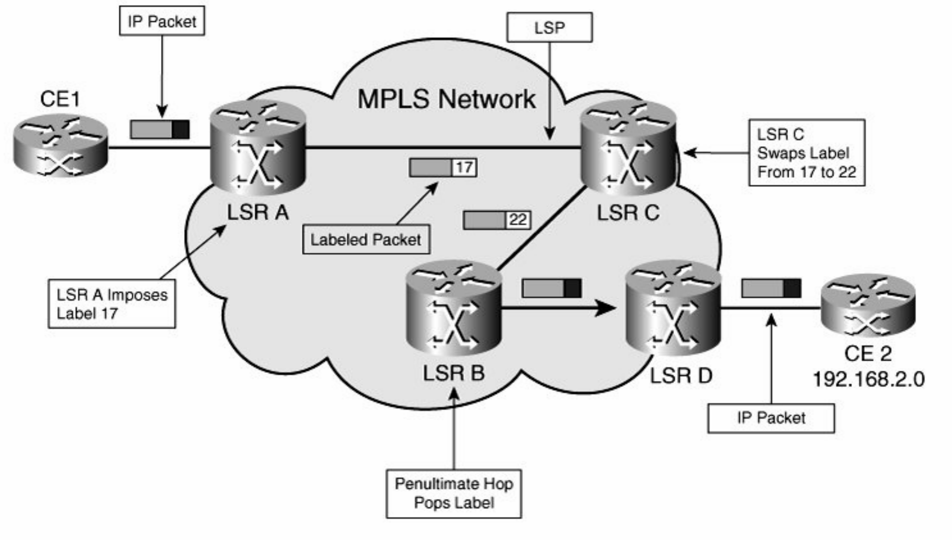


Fig (2-6).MPLS Forwarding

As a packet flows across the network shown in mk:@MSITStore:F:\ \Idol\M.Sc-CS%20IDOL%20COURCE%20WRITING%20CONTENT%20FOR%20ANC\Network%20Virtualization%20by%20Victor%20Moreno,%20Kumar%20Reddy%20(z-lib.org).chm::/1587052482/ch04lev1sec2.html - ch04fig10, it is processed by each hop as follows:

1. At the edge of the network, as shown in mk:@MSITStore:F:\ \Idol\M.Sc-CS%20IDOL%20COURCE%20WRITING%20CONTENT%20FOR%20ANC\Network%20Virtualization%20by%20Victor%20Moreno,%20Kumar%20Reddy%20(z-lib.org).chm::/1587052482/ch04lev1sec2.html - ch04fig10, edge LSR A classifies a packet to its FEC and assigns (or imposes) label 17 to the packet. A label is of local significance on that interface just like an ATM VPI/VCI or a Frame Relay DLCI.

2. In the core, LSRs, such as LSR C and LSR B, swap label values. LSR C removes the old label, 17 in the example shown in mk:@MSITStore:F:\ \Idol\M.Sc-CS%20IDOL%20COURCE%20WRITING%20CONTENT%20FOR%20ANC\Network%20Virtualization%20by%20Victor%20Moreno,%20Kumar%20Reddy%20(z-lib.org).chm::/1587052482/ch04lev1sec2.html - ch04fig10, and imposes the new one, 22. The values of the ingress label and interface are used to find the values of the egress label and interface.

   Note

   Not all MPLS forwarding modes use incoming interface. Frame mode, used in certain L2VPN services, just uses the incoming label as the same label value is advertised to all peers

3. LSR B, as the second-last hop in the MPLS network, removes the outermost label from the label stack, which is called penultimate hop popping (PHP). So, packets arrive at edge LSR D without any label, and standard IP routing is used to forward the packet. The process of

removing a label is also called disposition. PHP avoids recursive lookups on edge LSR D.

4. After the label is removed, the packet is forwarded using standard IP routing.

Now the difference with standard IP forwarding should be clearer. FEC classification is done when a packet enters the MPLS network, not at every hop. An LSR needs to look only at the packet's label to know which outgoing interface to use. There can be different labels on an LSR for the same IP destination. Saying the same thing in a different way, there can be multiple LSPs for the same destination.

A key point to understand is that the control plane is identical in both the IP and MPLS cases. LSRs use IP routing protocols to build routing tables, just as routers do. An LSR then goes the extra step of assigning labels for each destination in the routing table and advertising the label/FEC mapping to adjacent LSRs. ATM switches can also be LSRs. They run IP routing protocols, just as a router LSR does, but label switch cells rather than packets.

What is missing from this description is how label information is propagated around the network. How does LSR A in mk:@MSITStore:F:\ \Idol\M.Sc-CS%20IDOL%20COURCE%20WRITING%20CONTENT%20FOR%20ANC\Network%20Virtualization%20by%20Victor%20Moreno,%20Kumar%20Reddy%20(z-lib.org).chm::/1587052482/ch04lev1sec2.html - ch04fig10 know what label to use? MPLS networks use a variety of signaling protocols to distribute labels:

- LDP Used in all MPLS networks
- iBGP Used for L3 VPN service
- RSVP Used for Traffic Engineering
- Directed LDP Used for L2VPN service, such as VPLS

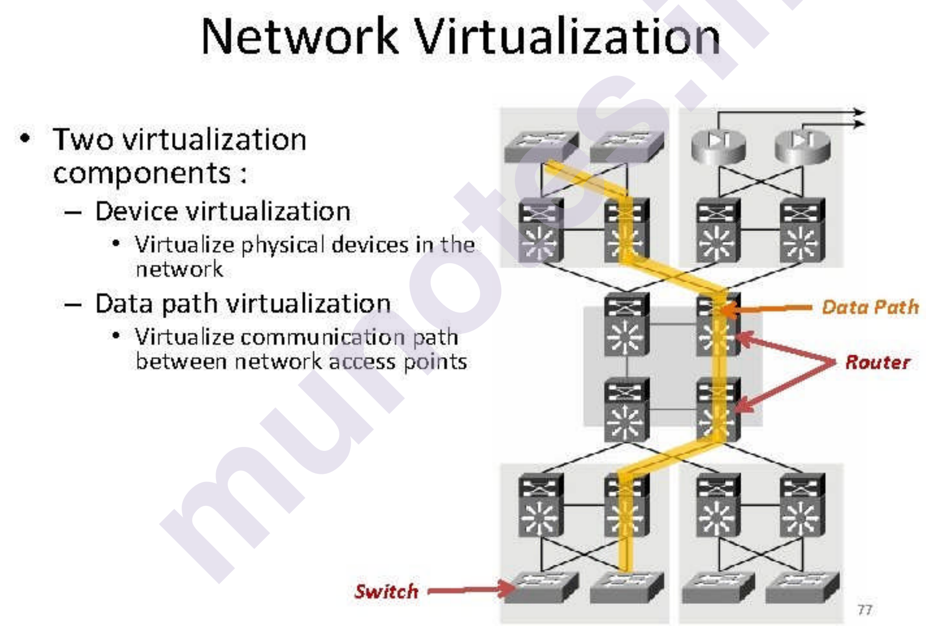
Label Distribution Protocol (LDP), which runs over tcp/646, is used in all MPLS networks to distribute labels for all prefixes in the nodes routing table. Referring again to mk:@MSITStore:F:\ \Idol\M.Sc-CS%20IDOL%20COURCE%20WRITING%20CONTENT%20FOR%20ANC\Network%20Virtualization%20by%20Victor%20Moreno,%20Kumar%20Reddy%20(z-lib.org).chm::/1587052482/ch04lev1sec2.html - ch04fig10, LSR D and LSR B would bring up a LDP session (LSR B would have another session with LSR C and so forth). LSR D is connected to the customer 192.168.2.0/24 network and advertises this prefix to all its routing peers. LSR D also sends a label to LSR B for the 192.168.2.0 network. When LSR B's routing protocol converges and it sees 192.168.2.0 as reachable, it sends label 22 to LSR C. This process continues until LSR A receives a label from LSR C.

The complete end-to-end set of labels from LSR A to LSR D form an LSP. An LSP is unidirectional. There is another LSP, identified by a different set of labels, for return traffic from LSR D to LSR A.

Understand that two operations must complete for the LSP from LSR A to 192.168.2.0 to be functional:

- The backbone routing protocol must converge so that LSR A has a route to 192.168.2.0.
- LDP must converge so that labels are propagated across the network.

Fig (2-6) does not show a numeric value for the label between LSR B and LSR D. In fact, as already discussed, the packet on this link has no label at all, because of PHP. Nevertheless, LSR D does still advertise a special value in LDP, called an implicit null (which has a reserved value of 3), so that LSR B performs PHP.



Network Virtualization

- Two virtualization components :
  - Device virtualization
    - Virtualize physical devices in the network
  - Data path virtualization
    - Virtualize communication path between network access points

Fig-(2-7) Network Virtualization

Above diagram simply shows the concept of Data path virtualization which comes under network virtualization technology. It provides virtualize communication path between network access points as shown in the above diagram.

Data-Path Virtualization Summary

We presented several different protocols that can be used for data-path virtualization. Two of them are suitable for Layer 2 traffic only: 802.1q, which is configured on each hop, and L2TPv3 which is configured end to end. IPsec is suitable for IP transport. Finally, GRE and MPLS LSPs can be used for either Layer 2 or Layer 3. GRE is another IP tunnel protocol, configured only on endpoints. MPLS creates a new forwarding path and is configured on all hops in a network.

Control-Plane Virtualization

Refers to all the protocols, databases, and tables necessary to make forwarding decisions and maintain a functional network topology free of loops or unintended blackholes. This plane could be said to draw a clear picture of the topology for the network device. A virtualized device must posses a unique picture of each VN it is to handle, hence the requirement to virtualize the control-plane components.

Data-path virtualization essentially creates multiple separate logical networks over a single, shared physical topology. To move packets across these VNs, you need to need a routing protocol.

The most familiar virtualized control plane is probably Per VLAN Spanning Tree (PVST), which has a separate spanning-tree instance for each VLAN running on a switch. Even through PVST has been around longer than the term virtualization, it illustrates the central point we are making here very crisply. Different logical networks have different topologies and, therefore, different optimal paths. Switches have to run different spanning-tree calculations for each such network.

The remainder of this section deals with extensions to routing protocols to allow them to run with multiple routing instances. However, we will return to the topic of control-plane virtualization, because many different router and switch functions, such as NetFlow, DHCP, RADIUS, and so on, need to receive the same treatment and become VRF aware.

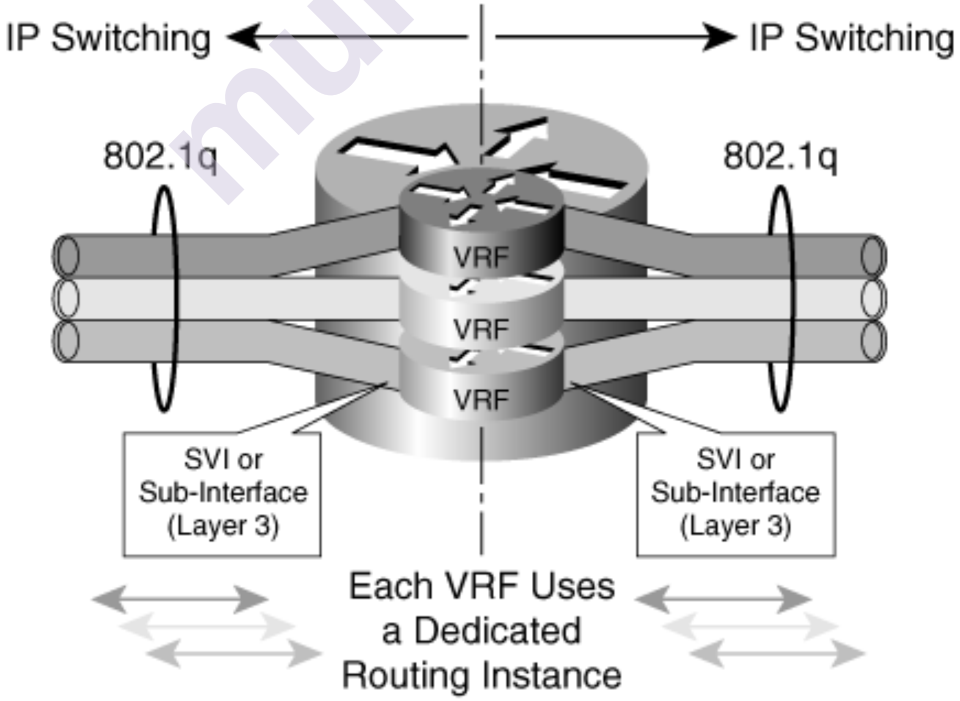

IP Switching
IP Switching
802.1q
802.1q
VRF
VRF
VRF
SVI or Sub-Interface (Layer 3)
SVI or Sub-Interface (Layer 3)
Each VRF Uses a Dedicated Routing Instance

Fig-(2-7)

VRF-Aware Routing

Cisco's major interior gateway protocol (IGP) routing protocol implementations are VRF aware. This means that they understand that certain routes may be placed only in certain routing tables. The routing protocols manage this by peering within a constrained topology, where a routing protocol instance in a VRF peers with other instances in the same VN. No special information is added to the route advertisements to identify VRF names, so routing instances must communicate over private links.

With some protocols (for example, BGP), a single routing instance can manage multiple VRF tables; with others (for example, OSPF), a different routing process runs for every VRF. Remember that in both cases, every VRF requires a route optimization calculation, so increasing the number of VRFs does have a computational impact on a network device.

Multi-Topology Routing

Multi-Topology Routing (MTR) is a recent innovation at Cisco. As the name suggests, it creates multiple routing topologies across a shared, common infrastructure. However, MTR does not try to be yet another VPN solution. Instead, it creates paths through a network that you can map to different applications or classes of applications, with the understanding that, by separating traffic in this way, you can provide better performance characteristics to certain critical applications.

MTR bases its operation on the creation of separate RIBs and FIBs for each topology. The separate RIBs and FIBs are created within a common address space. Thus, MTR creates smaller topologies that are a subset of the full topology (also known as the base topology). The main difference between MTR and a VPN technology is that, with MTR, a single address space is tailored into many topologies that could overlap; whereas VPNs create totally separate and independent address spaces.

Thus, MTR must carry out two distinct functions:

- At the control planeColor the routing updates, so that the different topology RIBs are populated accordingly. Based on these RIBs, the corresponding FIBs are to be written.
- At the forwarding plane Identify the topology to which each packet belongs and use the correct FIB to forward the packet.

At each hop, there will be a set of prefixes and routes in the RIB for each topology. The contents of these RIBs are dynamically updated by routing protocol colored updates. Based on this RIB information, a separate FIB is built for each topology.

To forward traffic over different topologies, the router looks for a code point in each packet and chooses an FIB based on this code point. A first implementation of MTR uses differentiated services (DiffServ) code point (DSCP) as such a code point, but other code points could be used by future implementations. The DSCP value is used as a pointer to the correct forwarding table, and the packet's destination address is used to make a forwarding decision based on the information in the topology's FIB. MTR uses the terminology of color to refer to separate topologies. So, a RED value in a packet's DSCP field is recognized by the router, which will forward the packet using the RED forwarding table (FIB).

MTR must run contiguously across a network, and the color mappings must be consistent (that is, you cannot use DSCP X as Green on one hop but as Red on the next). MTR does not allow you to double dip: If the destination route is not in the routing table of the color a packet is using, the packet can either be dropped or forwarded over the base topology there are no lookups in "backup topologies" other than the base topology (which is equivalent to regular routing).

MTR does not change how routing works; it just runs across multiple topologies (using a single process with colored updates).

Control-Plane Virtualization Summary

Control-plane virtualization refers to adaptations made to routing protocols to be able to operate on virtualized devices. We concentrated on per-VRF routing because that is the main tool we use for design. However, VRs and LRs also run separate routing instances in a similar manner to the one shown here. In all cases, there are no changes to the protocol "on the wire." MTR is an interesting new development that can also be categorized in the virtualized control-plane bucket.

Routing Protocols

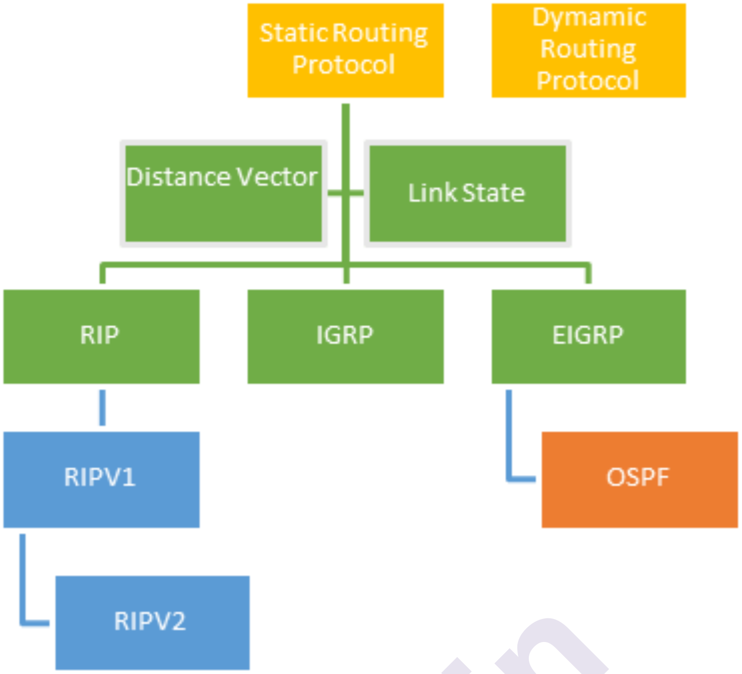
Routing Protocols are the set of defined rules used by the routers to communicate between source & destination. They do not move the information to the source to a destination, but only update the routing table that contains the information.

Network Router protocols help you to specify way routers communicate with each other. It allows the network to select routes between any two nodes on a computer network.

Types of Routing Protocols

There are mainly two types of Network Routing Protocols

- Static
- Dynamic



Routing Protocols

Static Routing Protocols

Static routing protocols are used when an administrator manually assigns the path from source to the destination network. It offers more security to the network.

Advantages

- No overhead on router CPU.
- No unused bandwidth between links.
- Only the administrator is able to add routes

Disadvantages

- The administrator must know how each router is connected.
- Not an ideal option for large networks as it is time intensive.
- Whenever link fails all the network goes down which is not feasible in small networks.

Dynamic Routing Protocols

Dynamic routing protocols are another important type of routing protocol. It helps routers to add information to their routing tables from connected routers automatically. These types of protocols also send out topology updates whenever the network changes' topological structure.

Advantage:

- Easier to configure even on larger networks.
- It will be dynamically able to choose a different route in case if a link goes down.
- It helps you to do load balancing between multiple links.

Disadvantage:

- Updates are shared between routers, so it consumes bandwidth.
- Routing protocols put an additional load on router CPU or RAM.

Distance Vector Routing Protocol (DVR)

Distance Vector Protocols advertise their routing table to every directly connected neighbor at specific time intervals using lots of bandwidths and slow converge.

In the Distance Vector routing protocol, when a route becomes unavailable, all routing tables need to be updated with new information.

Advantages:

- Updates of the network are exchanged periodically, and it is always broadcast.
- This protocol always trusts route on routing information received from neighbor routers.

Disadvantages:

- As the routing information are exchanged periodically, unnecessary traffic is generated, which consumes available bandwidth.

Internet Routing Protocols:

The following are types of protocols which help data packets find their way across the Internet:

Routing Information Protocol (RIP)

RIP is used in both LAN and WAN Networks**.** It also runs on the Application layer of the OSI model. The full form of RIP is the Routing Information Protocol. Two versions of RIP are

1. RIPv1
2. RIPv2

The original version or RIPv1 helps you determine network paths based on the IP destination and the hop count journey. RIPv1 also interacts with the network by broadcasting its IP table to all routers connected with the network.

RIPv2 is a little more sophisticated as it sends its routing table on to a multicast address.

Interior Gateway Protocol (IGP)

IGRP is a subtype of the distance-vector interior gateway protocol developed by CISCO. It is introduced to overcome RIP limitations. The metrics used are load, bandwidth, delay, MTU, and reliability. It is widely used by routers to exchange routing data within an autonomous system.

This type of routing protocol is the best for larger network size as itbroadcasts after every 90 seconds, and it has a maximum hop count of 255**.** It helps you to sustain larger networks compared to RIP. IGRP is also widely used as it is resistant to routing loop because it updates itself automatically when route changes occur within the specific network. It is also given an option to load balance traffic across equal or unequal metric cost paths.

Link State Routing Protocol

Link State Protocols take a unique approach to search the best routing path. In this protocol, the route is calculated based on the speed of the path to the destination and the cost of resources.

Routing protocol tables:

Link state routing protocol maintains below given three tables:

- **Neighbor table:** This table contains information about the neighbors of the router only. For example, adjacency has been formed.
- **Topology table:** This table stores information about the whole topology. For example, it contains both the best and backup routes to a particular advertised network.
- **Routing table:** This type of table contains all the best routes to the advertised network.

Advantages:

- This protocol maintains separate tables for both the best route and the backup routes, so it has more knowledge of the inter-network than any other distance vector routing protocol.
- Concept of triggered updates are used, so it does not consume any unnecessary bandwidth.
- Partial updates will be triggered when there is a topology change, so it does not need to update where the whole routing table is exchanged.

Exterior Gateway Protocol (EGP)

EGP is a protocol used to exchange data between gateway hosts that are neighbors with each other within autonomous systems. This routing protocol offers a forum for routers to share information across different domains. The full form for EGP is the Exterior Gateway Protocol. EGP protocol includes known routers, network addresses, route costs, or neighboring devices.

Enhanced Interior Gateway Routing Protocol (EIGRP)

EIGRP is a hybrid routing protocol that provides routing protocols, distance vector, and link-state routing protocols. The full form routing protocol EIGRP is Enhanced Interior Gateway Routing Protocol. It will route the same protocols that IGRP routes using the same composite metrics as IGRP, which helps the network select the best path destination.

Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) protocol is a link-state IGP tailor-made for IP networks using the Shortest Path First (SPF) method.

OSPF routing allows you to maintain databases detailing information about the surrounding topology of the network. It also uses the Dijkstra algorithm (Shortest path algorithm) to recalculate network paths when its topology changes. This protocol is also very secure, as it can authenticate protocol changes to keep data secure.

Here are some main difference between these Distance Vector and Link State routing protocols:

Distance Vector	**Link State**
Distance Vector protocol sends the entire routing table.	Link State protocol sends only link-state information.
It is susceptible to routing loops.	It is less susceptible to routing loops.
Updates are sometimes sent using broadcast.	Uses only multicast method for routing updates.
It is simple to configure.	It is hard to configure this routing protocol.
Does not know network topology.	Know the entire topology.
Example RIP, IGRP.	Examples: OSPF IS-IS.

Intermediate System-to-Intermediate System (IS-IS)

ISIS CISCO routing protocol is used on the Internet to send IP routing information. It consists of a range of components, including end systems, intermediate systems, areas, and domains.

The full form of ISIS is Intermediate System-to-Intermediate System. Under the IS-IS protocol, routers are organized into groups called areas. Multiple areas are grouped to make form a domain.

Border Gateway Protocol (BGP)

BGP is the last routing protocol of the Internet, which is classified as a DPVP (distance path vector protocol). The full form of BGP is the Border Gateway Protocol.

This type of routing protocol sends updated router table data when changes are made. Therefore, there is no auto-discovery of topology changes, which means that the user needs to configure BGP manually.

What is the purpose of Routing Protocols?

Routing protocols are required for the following reasons:

- Allows optimal path selection
- Offers loop-free routing
- Fast convergence
- Minimize update traffic
- Easy to configure
- Adapts to changes
- Scales to a large size
- Compatible with existing hosts and routers
- Supports variable length

Classful Vs. Classless Routing Protocols

Here are some main difference between these routing protocols:

Classful Routing Protocols	**Classless Routing Protocols**
Classful routing protocols never send subnet mask detail during routing updates.	Classless routing protocols can send IP subnet mask information while doing routing updates.
RIPv1andIGRP are classful protocols. These two are classful protocols as they do not include subnet mask information.	RIPv2, OSPF, EIGRP, and IS-IS are all types of class routing protocols which has subnet mask information within updates.

5.5 SUMMARY:

Features	**RIP V1**	**RIP V2**	**IGRP**	**OSPF**	**EIGRP**
Classful/ Classless	Classful	Classless	Classful	Classless	Classless
Metric	Hop	Hop	Composite Bandwidth, Delay.	Bandwidth	Composite, Bandwidth, Delay.
Periodic	30 seconds	30 seconds	90 seconds	None	30 seconds
Advertising Address	255.255.255.255.255	223.0.0.9	255.255.255.255.255	224.0.0.5 224.0.0.6	224.0.0.10
Category	Distance Vector	Distance Vector	Distance Vector	Link State	Hybrid
Default	120	120	200	110	170

Features	**RIP V1**	**RIP V2**	**IGRP**	**OSPF**	**EIGRP**
Distance					

❖❖❖❖

Module - III

6

ADHOC NETWORKING

Unit Structure

6.1 INTRODUCTION

Recent trends in compact computing and wireless technologies are expansion of ad hoc network. Ad hoc network consists of versatile flat forms which are free to move expeditiously. Ad hoc networks are multi-hop network that use wireless communication for transmission without any fixed infrastructure.

Adhoc network is an autonomous system node connected with wireless link The node in the ad hoc network communicates with other node without any physical representation. The nodes in the ad hoc organization instantly form the network whenever the communication is established. Each node in the network communicates with other node using radio waves. The entire network is distributed and nodes are collaborated with each other without fixed station access point (AP) or base station. An ad hoc network is local area network that builds an automatic connection to the nodes in the network

In the Windows operating system, ad hoc is a communication mode (setting) that allows computers to directly communicate with each other without a router. Wireless mobile ad hoc networks are self-configuring, dynamic networks in which nodes are free to move.

In the Windows operating system, ad hoc is a communication mode (setting) that allows computers to directly communicate with each other without a router. Wireless mobile ad hoc networks are self-configuring, dynamic networks in which nodes are free to move.

Following figures shows the difference between infrastructure based wireless network and Ad hoc wireless network.

Fig. 1. Infrastructure based wireless networks.

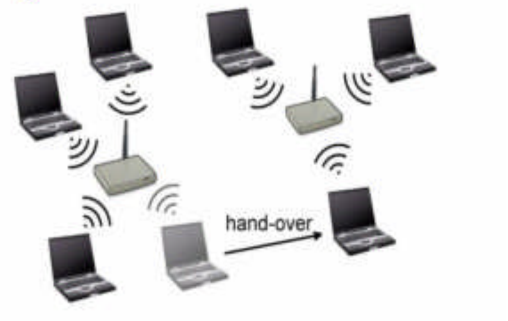
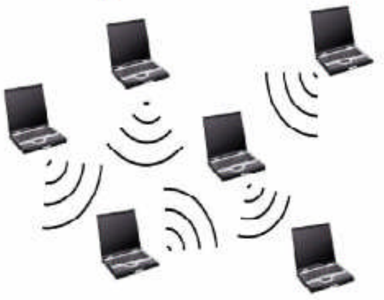


Fig. 2. Ad hoc wireless.



As you can see in infrastructure based wireless network there is using of a network device called a router of a switch but those network devices are absent into Ad hoc wireless network.

6.2 APPLICATION OF MANET

Mobile ad hoc network (**MANET**) is a decentralized type of wireless network.The network is said to be ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks.

With the increased number of lightweight devices as well as evolution in wireless communication, the ad hoc networking technology is gaining effort with the increasing number of widespread applications.

Ad hoc networking can be used anytime, anywhere with limited or no communication infrastructure. The preceding infrastructure is fancy or annoying to use. The ad hoc network architecture can be used in real time business applications, corporate companies to increase the productivity and profit.

The ad hoc networks can be classified according to their application as Mobile Ad hoc Network (MANET) which is a self-arranging infrastructure less network of mobile devices communicated through wireless link.

Vehicular Ad hoc Network (VANET) uses travelling cars as nodes in a network to create a mobile network. Wireless Sensor Network (WSN) consists of autonomous sensors to control the environmental actions. The importance of ad hoc network has been highlighted in many fields which are described below:

Military arena: An ad hoc networking will allow the military battleground to maintain an information network among the soldiers, vehicles and headquarters.

Provincial level: Ad hoc networks can build instant link between multimedia network using notebook computers or palmtop computers to spread and share information among participants (e.g. Conferences).

Personal area network: A personal area network is a short range, localized network where nodes are usually associated with a given range.

Industry sector: Ad hoc network is widely used for commercial applications. Ad hoc network can also be used in emergency situation such as disaster relief. The rapid development of non-existing infrastructure makes the ad hoc network easily to be used in emergency situation.

Bluetooth: Bluetooth can provide short range communication between the nodes such as a laptop and mobile phone.

6.3 CHALLENGES:

The ad hoc networks are self-forming, self-maintaining, self-healing architecture. The challenges are, no fixed access point, dynamic network topology, contrary environment and irregular connectivity. Ad hoc network immediately forms and accommodate the modification and limited power. Finally, ad hoc have no trusted centralized authority. Due to the dynamic changing property, the ad hoc faces some challenges which are listed in the below sections.

Quality of Service (QoS) The ad hoc network is dynamically creating the organization whenever the node wants to communicate with their neighbour node. Due the dynamic changing topology in ad hoc network, providing QoS is a tedious task.

QoS are essential because of rapid development in mobile technology and real time applications like multimedia, voice. Providing QoS in ad hoc network is necessary to maintain best-effort-of service.

The QoS metric are bandwidth, latency, jitter and delivery guarantee. The bandwidth is used to denote the data rate carried in the network. Latency ensures the delay occur from origin to target. Jitter denotes the variation of delay. Reliability demonstrate the percentage of deny to access the network service.

Wireless channels are varying rapidly and it severely affects the multi-hop flows. In ad hoc networks, the peer-to-peer channel quality may alter rapidly. So, the link quality may affect the peer-to-peer QoS metrics in the multi-hop path.

Limited Bandwidth

The wireless networks have a limited bandwidth in comparison to the wired networks. Wireless link has lower capacity as compare to infrastructure networks. The effect of fading, multiple accesses, interference condition is very low in ADHOC networks in comparison to maximum radio transmission rate.

Dynamic topology

Due to dynamic topology the nodes has less trust between them. I some settlement are found between the nodes then it also makes trust level questionable.

High Routing

In ADHOC networks due to dynamic topology some nodes change their position which affects the routing table.

Problem of Hidden terminal

The Collision of the packets are held due to the transmission of packets by those nodes which are not in the direct transmission range of sender side but are in range of receiver side.

Transmission error and packet loss

By increasing in collisions, hidden terminals, interference, uni-directional links and by the mobility of nodes frequent path breaks a higher packet loss has been faced by ADHOC networks.

Mobility

Due to the dynamic behaviour and changes in the network topology by the movement of the nodes. ADHOC networks faces path breaks and it also changes in the route frequently.

Security threats

New security challenges bring by Ad hoc networks due to its wireless nature. In Ad hoc networks or wireless networks, the trust management between the nodes leads to the numerous security attacks.

Routing in Ad hoc networks

In ad hoc networks, nodes are not familiar with the topology of their networks. Instead, they have to discover it: typically, a new node announces its presence and listens for announcements broadcast by its neighbours. Each node learns about others nearby and how to reach them, and may announce that it too can reach them.

There are basic 4 types of routing in ad hoc network is present listed below.

1) **Table-driven (proactive) routing**
2) **On-demand (reactive) routing**
3) **Hybrid (both proactive and reactive) routing**
4) **Hierarchical routing protocols**

1) Table-driven (proactive) routing

- This type of protocols maintains fresh lists of destinations and their routes by periodically distributing routing tables throughout the network.
- In proactive routing (table-driven routing), the routing tables are created before packets are sent – Link-state (e.g. OSPF) – Distance-vector (e.g. RIP)
- Each node knows the routes to all other nodes in the network
- Problems in Ad-Hoc networks
- Maintenance of routing tables requires much bandwidth
- Dynamic topology ˇ much of the routing information is never used ˇWaste of capacity
- Flat topology ˇNo aggregation

2) On-demand (reactive) routing

- This type of protocol finds a route on demand by flooding the network with Route Request packets.
- In reactive routing the routes are created when needed.
- Before a packet is sent, a route discovery is performed.
- The results are stored in a cache.
- When intermediate nodes move, a route repair is required.
- Advantages – Only required routes are maintained
- Disadvantages
- Delay before the first packet can be sent
- Route discovery usually involves flooding

3) Hybrid (both proactive and reactive) routing

- This type of protocol combines the advantages of proactive and reactive routing.
- The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding.
- The choice of one or the other method requires predetermination for typical cases.

4) Hierarchical routing protocols

- With this type of protocol, the choice of proactive and of reactive routing depends on the hierarchic level in which a node resides.
- The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding on the lower levels.
- The choice for one or the other method requires proper attributation for respective levels.

Routing Protocols: Topology based

An ad hoc wireless multi-hop network (AHWMNs) is a collection of mobile devices which form a communication network with no pre-existing wiring or infrastructure. Routing in AHWMNs is challenging since there is no central coordinator that manage routing decisions. AHWMN routing protocols are classified as topology-based, position-based.

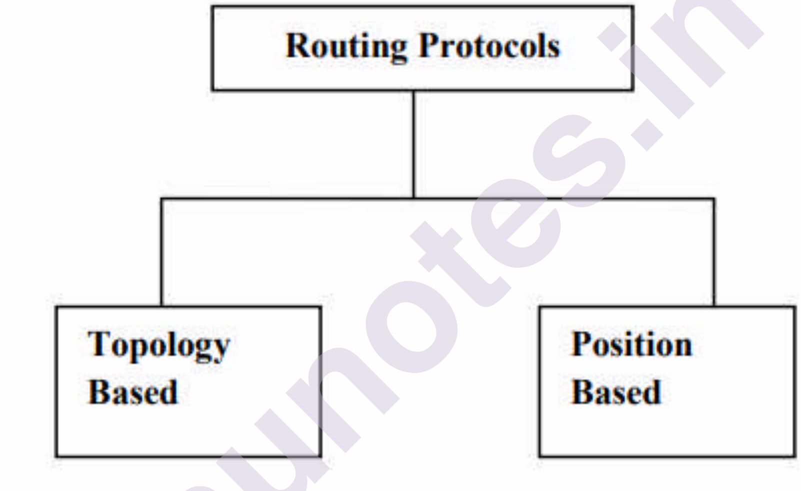


Fig. 3.Classification of routing protocol

Topology-based routing protocols use the information about the links that exist in the network to perform packet forwarding. They can be further divided into proactive, reactive and hybrid approaches.

Proactive algorithms employ classical routing strategies such as distance- vector routing (e.g. DSDV) or link-state routing (e.g. OLSR). They maintain routing information about the available paths in the network even if these paths are currently not used. The main drawback of these approaches is the maintenance of unusual path may occupy a significant part of the available bandwidth if the topology of the network changes frequently.

Reactive routing protocols such as AODV and DSR maintain only the routes that are currently in use and hence reduce the burden on the network. However, they still have some inherent limitations. First, since the routes are maintained only while in use, it is required to perform a route discovery before packets are exchanged between communication nodes. Second, even though route discovery is restricted to the routes currently in use, it may still generate a significant amount of network traffic when the topology of the network changes frequently.

Position-based routing algorithms eliminate some of the limitations of topology-based routing by using additional information. Position based routing based on idea that the source sends a message to the geographic location of destination instead of using the network address. Position based routing requires information about the physical position of participating nodes. Commonly, each node determines its own position through the use of Global Positioning System (GPS). Decisions made based on destination position and position of forwarding nodes neighbors. A location service is used by the sender of packet to determine the position of the destination and to include it in the packet destination address.

Greedy Perimeter Stateless Routing (GPSR) protocol is an efficient and scalable routing protocol in MANETs. In GPSR protocol, a node route the data packet using the locations of its one hop neighbors. When the node needs to send a data packet to destination node, it transmits the data packet to the neighbour who has the shortest distance to the destination node among all its neighbors within its transmission range. GPSR protocol uses two forwarding strategies to route the data packet to the destination. They are greedy forwarding and perimeter forwarding. GPSR makes greedy forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region. By keeping state only about the local topology, GPSR scales better in per-router state than shortest-path and ad-hoc routing protocols as the number of network destinations increases. Under mobility's frequent topology changes, GPSR can use local topology information to find correct new routes quickly.

In GPSR, packets are marked by their originator with their destinations' locations. As a result, a forwarding node can make a locally optimal, greedy choice in choosing a packet's next hop. Specifically, if a node knows its radio neighbors' positions, the locally optimal choice of next hop is the neighbor geographically closest to the packet's destination. Forwarding in this regime follows successively closer geographic hops, until the destination is reached.

Position based

Position based routing are as follows:

1) GFG

2) GOAFR

3) GOAFR+

4) AFR

5) OGPR

6) GPVFR

7) LAR

1) GFG

Nearly Stateless Routing with Guaranteed Delivery is schemes where nodes maintain only some local information to perform routing. The face routing and Greedy-Face-Greedy (GFG) schemes were described in. In order to ensure message delivery, the face routing (called perimeter algorithm) constructs a planar and connected so-called Gabriel sub graph of the unit graph, and then applies routing along the faces of the sub graph (e.g. by using the right hand rule) that intersect the line between the source and the destination. If a face is traversed using the right hand rule then a loop will be created, since a face will never exist. Forwarding in the right hand rule is performed using the directional approach.

To improve the efficiency of the algorithm in terms of routing performance, face routing can be combined with algorithms that usually find shorter routes, such as the greedy algorithm to yield GFG algorithm. Routing is mainly greedy, but if a mobile host fails to find a neighbor closer than itself to the destination, it switches the message from ‘greedy’ state to ‘face’ state.

2) GOAFR

A greedy routing approach is not only worth being considered due to its simplicity in both concept and implementation. Above all in dense networks such an algorithm can also be expected to end paths of good quality efficiently here, the straightforwardness of a greedy strategy contrasts highly the inexible exploration of faces inherent to face routing. For practical purposes it is inevitable to improve the performance of a face routing variant by leveraging the potential of a greedy approach. Such a combination of greedy routing and our OAFR algorithm forms Greedy Other Adaptive Face Routing GOAFR. In principle greedy routing is used as long as possible. Local minima potentially met under ways are escaped from by use of OAFR.

3) GOAFR+

The GOAFR+ algorithm is a combination of greedy routing and face routing. Whenever possible the algorithm tries to route greedily, that is by forwarding the message at each intermediate node to the neighbour located closest to the destination. Doing so, however, the algorithm can reach a local minimum with respect to the distance from destination that is a node um none of whose neighbours is located closer to destination than itself. In order to overcome such a local minimum, GOAFR+ applies a face routing technique, borrowing from the Face Routing algorithm.

Face Routing proceeds towards the destination by exploring the boundaries of the faces of a planarized network graph, employing the local right hand rule (in analogy to following the right hand wall in a maze).

Additionally the algorithm restricts itself to a searchable area occasionally being resized during algorithm execution.

4) AFR

The basis of this algorithm is formed by Face Routing. At the heart of Face Routing lies the exploration of the boundaries of faces in a planar graph, employing the local right hand rule (in analogy to following the right hand wall in a maze). On its way around a face, the algorithm keeps track of the points where it crosses the line connecting the source and the destination. Having completely surrounded a face, the algorithm returns to the one of these intersections lying closest to the destination, where it proceeds by exploring the next face closer to destination. If the source and the destination are connected, Face Routing always ends a path to the destination.

5) OGPR

OGPR is an efficient and scalable routing protocol, that inherits the well-known techniques for routing,
1) Greedy forwarding
2) Reactive route discovery
3) Source routing

In this, protocol source node utilizes the geographic topology information obtained during the location request phase to establish geographic paths to their respective destinations.

6) GPVFR

In this section we describe Greedy PVFR, a non-oblivious routing algorithm that does not require the participating nodes to have complete face information. GPVFR is designed as a tri-modal algorithm with the following modes

- Greedy: greedy forwarding using neighbour information,
- OPVFR: greedy forwarding using face information, and
- Perimeter: perimeter traversal (as in GPSR).

Under GPVFR, packets are first routed in Greedy mode. When greedy forwarding to an immediate neighbour fails, a node may find that it knows of another node along its planar faces that is nearer to the destination than itself.

7) LAR

The Location Aided Routing proposal does not define a location-based routing protocol but instead proposes the use of position information to enhance the route discovery phase of reactive ad hoc routing approaches. Reactive ad hoc routing protocols frequently use flooding as a means of route discovery.

Under the assumption that nodes have information about other nodes' positions, this position information can be used by LAR to restrict the flooding to a certain area. This is done in a fashion similar to that of the DREAM approach. When node S wants to establish a route to node D, S computes an expected zone for D based on available position information. If no such information is available LAR is reduced to simple flooding. If location information is available (e.g., from a route that was established earlier), a request zone is defined as the set of nodes that should forward the route discovery packet.

The request zone typically includes the expected zone. The first is a rectangular geographic region. In this case, nodes will forward the route discovery packet only if they are within that specific region. The second is defined by specifying (estimated) destination coordinates plus the distance to the destination. In this case, each forwarding node overwrites the distance field with its own current distance to the destination. A node is allowed to forward the packet again only if it is at most some δ (system parameter) farther away than the previous node.

6.4 BROADCASTING

Multicasting

In multicasting routing, the data are transmitted from one source to multiple destinations. Multicast protocols can be categorized into two types, namely tree-based multicast and mesh based multicast. The tree based multicast routing protocols utilize the network resource in efficient manner. Mesh based protocols are robust due to formation of many redundant paths between the nodes and in high packet delivery ratio.

Ad hoc multicast routing protocol (AMRoute): Xie et al. [75] developed AMRoute, with main design objective are: scalability and robustness. In ad hoc network with highly dynamic mobile nodes, the control packets overhead are high due to maintenance of multi cast tree.

Adaptive demand-driver multicast routing (ADMR): ADMR, on-demand multicast routing algorithm, developed by Jetchera and Johnson [76]. This protocol does not support any non on-demand components. ADMR, uses a source based forwarding trees and monitors the traffic pattern and rate of the source. ADMR navigates back to the normal mode, when the mobility of the node is reduced.

Differential destination multicast (DDM): Ji and Corson [77] proposed the DDM algorithm. DDM has two important characteristics features: 1. the sender node will have full control over the members of group nodes. 2. Source node, encodes the address within each data packets header on an in-band fashion.

Dynamic core based multicast routing (DCMP): Das et al. [78] proposed DCMP source initiated multicast protocol with an objective to increase the scalability and efficiency as well as to decrease the overhead. In this protocol the source as been classified into active, core active and passive. A core active source can support up to maximum of MaxPassSize passive resource and the hop distance between them is limited by the MaxHop parameter.

AdhocQoS multicasting (AQM): AQM protocols developed by Bur and Ersoy [79]. In this protocol QoS of the neighboring node monitored and maintained as well as used for efficient multicast routing. Node announces the QoS status during the session initiation phase to join a session, the nodes executes request-reply–reverse procedure, ensures the QoS information is updated and a possible route is chosen session is initiated by a session initiator node.

Content based multicast (CBM): CBM developed by Zhou and Singh [80]. In CBM the nodes collect information about threats and resource at a time period t and distance d away from the location of the node.

Energy efficient multicast routing: Li et al. [81] proposed an energy efficient multicast routing protocol. The authors constructed a weighted network graph by considering the transmission power of each node as a weight between edges. Each node has only information regarding their neighbors. The objective of minimum energy multicast (MEM), problem is to develop the multicast tree with a minimum total energy cost. In this approach, multicast tree is formed by nodes within the highest energy efficiency.

QoS multicast routing protocols for clustering mobile ad hoc networks (QMRPCAH): QMRPCAH, QoS aware multicast routing protocol for clustered ad hoc network was developed by Layuan and Chunlin [82]. It enhances scalability and flexibility.

Epidemic-based reliable and adaptive multicast for mobile ad hoc networks (Eramobile): Eramobile, highly reliable and an adaptive multicast protocol proposed by Ozkasap et al. [83]. In this protocol bio-inspired epidemic methods are utilized in multicast operation in order to support dynamic and topology changes due the unpredictable mobility of the nodes in the network. Table 7 illustrates the comparative analysis of multicast routing protocol.

Geocasting

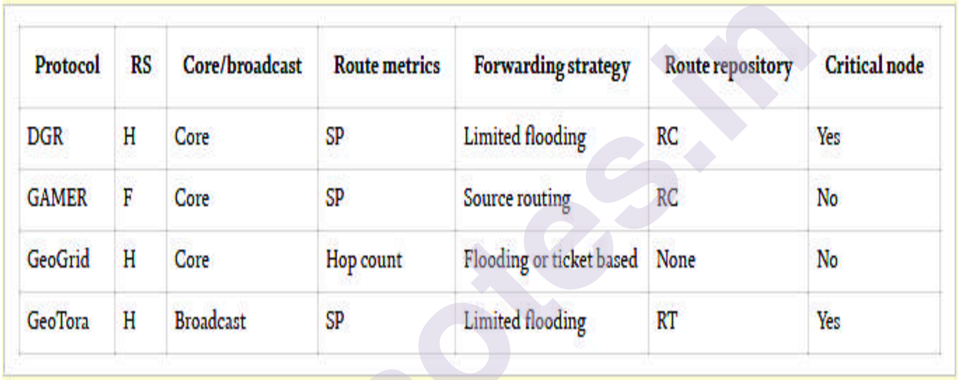
Geocast routing protocols have the combined features of both geographical and multicast routing protocols. The major advantage of Geocast routing protocols are performance improvement and minimizing the control overhead.

Geocasting in mobile ad hoc networks (GeoTORA): Ko and Vaidya [8] proposed the GeoTORA protocol, is based upon the unicast TORA routing protocol.

Geocast protocol for mobile ad hoc network based on GRID (GEOGRID): GeoGRID routing protocol was developed by Liao et al. [7] GeoGrid extends on the unicasting routing protocol GRID. GeoGRID exploit location information in route discovery to define the forwarding zone or geographical area.

Direction guided routing (DGR): An and Papavassilliou [94] designed DGR algorithm based on clustering mechanism. In DGR, the nodes in the network are grouped into clusters and the cluster head is elected using the techniques such a mobile clustering algorithm (MCA).

Geocast adaptive mesh environment for routing (GAMER): GAMER protocol developed by Camp and Liu [95] is based on the mobility nature of nodes. This protocol exploits the mesh creation approach. Table 9 illustrates the Geocast routing protocols comparative analysis.



Protocol	RS	Core/broadcast	Route metrics	Forwarding strategy	Route repository	Critical node
DGR	H	Core	SP	Limited flooding	RC	Yes
GAMER	F	Core	SP	Source routing	RC	No
GeoGrid	H	Core	Hop count	Flooding or ticket based	None	No
GeoTora	H	Broadcast	SP	Limited flooding	RT	Yes

RS = routing structure; H = hierarchical; F = flat; SP = shortest path; RC = route cache; RT = route table.

Wireless Lan

Wireless LAN stands for **Wireless Local Area Network**. It is also called LAWN (**Local Area Wireless Network**). WLAN is one in which a mobile user can connect to a Local Area Network (LAN) through a wireless connection.

The IEEE 802.11 group of standards defines the technologies for wireless LANs. For path sharing, 802.11 standard uses the Ethernet protocol and CSMA/CA (carrier sense multiple access with collision avoidance). It also uses an encryption method i.e. wired equivalent privacy algorithm.

Wireless LANs provide high speed data communication in small areas such as building or an office. WLANs allow users to move around in a confined area while they are still connected to the network.

In some instance wireless LAN technology is used to save costs and avoid laying cable, while in other cases, it is the only option for providing high-speed internet access to the public. Whatever the reason, wireless solutions are popping up everywhere.

Examples of WLANs that are available today are NCR's waveLAN and Motorola's ALTAIR.

Advantages of WLANs

- **Flexibility:** Within radio coverage, nodes can communicate without further restriction. Radio waves can penetrate walls, senders and receivers can be placed anywhere (also non-visible, e.g., within devices, in walls etc.).
- **Planning:** Only wireless ad-hoc networks allow for communication without previous planning, any wired network needs wiring plans.
- **Design:** Wireless networks allow for the design of independent, small devices which can for example be put into a pocket. Cables not only restrict users but also designers of small notepads, PDAs, etc.
- **Robustness:** Wireless networks can handle disasters, e.g., earthquakes, flood etc. whereas, networks requiring a wired infrastructure will usually break down completely in disasters.
- **Cost:** The cost of installing and maintaining a wireless LAN is on average lower than the cost of installing and maintaining a traditional wired LAN, for two reasons. First, after providing wireless access to the wireless network via an access point for the first user, adding additional users to a network will not increase the cost. And second, wireless LAN eliminates the direct costs of cabling and the labor associated with installing and repairing it.
- **Ease of Use:** Wireless LAN is easy to use and the users need very little new information to take advantage of WLANs.

Disadvantages of WLANs

- **Quality of Services:** Quality of wireless LAN is typically lower than wired networks. The main reason for this is the lower bandwidth due to limitations is radio transmission, higher error rates due to interference and higher delay/delay variation due to extensive error correction and detection mechanisms.
- **Proprietary Solutions:** Due to slow standardization procedures, many companies have come up with proprietary solutions offering standardization functionality plus many enhanced features. Most components today adhere to the basic standards IEEE 802.11a or 802.11b.
- **Restrictions:** Several govt. and non-govt. institutions world-wide regulate the operation and restrict frequencies to minimize interference.

- **Global operation:** Wireless LAN products are sold in all countries so, national and international frequency regulations have to be considered.
- **Low Power:** Devices communicating via a wireless LAN are typically power consuming, also wireless devices running on battery power. Whereas the LAN design should take this into account and implement special power saving modes and power management functions.
- **License free operation:** LAN operators don't want to apply for a special license to be able to use the product. The equipment must operate in a license free band, such as the 2.4 GHz ISM band.
- **Robust transmission technology:** If wireless LAN uses radio transmission, many other electrical devices can interfere with them (such as vacuum cleaner, train engines, hair dryers, etc.). Wireless LAN transceivers cannot be adjusted for perfect transmission is a standard office or production environment.

6.5 FUNDAMENTALS OF WLANS

1. HiperLAN

- HiperLAN stands for High performance LAN. While all of the previous technologies have been designed specifically for an adhoc environment, HiperLAN is derived from traditional LAN environments and can support multimedia data and asynchronous data effectively at high rates (23.5 Mbps).
- A LAN extension via access points can be implemented using standard features of the HiperLAN/1 specification. However, HiperLAN does not necessarily require any type of access point infrastructure for its operation.
- HiperLAN was started in 1992, and standards were published in 1995. It employs the 5.15GHz and 17.1 GHz frequency bands and has a data rate of 23.5 Mbps with coverage of 50m and mobility< 10 m/s.
- It supports a packet-oriented structure, which can be used for networks with or without a central control (BS-MS and ad-hoc). It supports 25 audio connections at 32kbps with a maximum latency of 10 ms, one video connection of 2 Mbps with 100 ms latency, and a data rate of 13.4 Mbps.
- HiperLAN/1 is specifically designed to support adhoc computing for multimedia systems, where there is no requirement to deploy a centralized infrastructure. It effectively supports MPEG or other state of the art real time digital audio and video standards.
- The HiperLAN/1 MAC is compatible with the standard MAC service interface, enabling support for existing applications to remain unchanged.

- HiperLAN 2 has been specifically developed to have a wired infrastructure, providing short-range wireless access to wired networks such as IP and ATM.

The two main differences between HiperLAN types 1 and 2 are as follows:

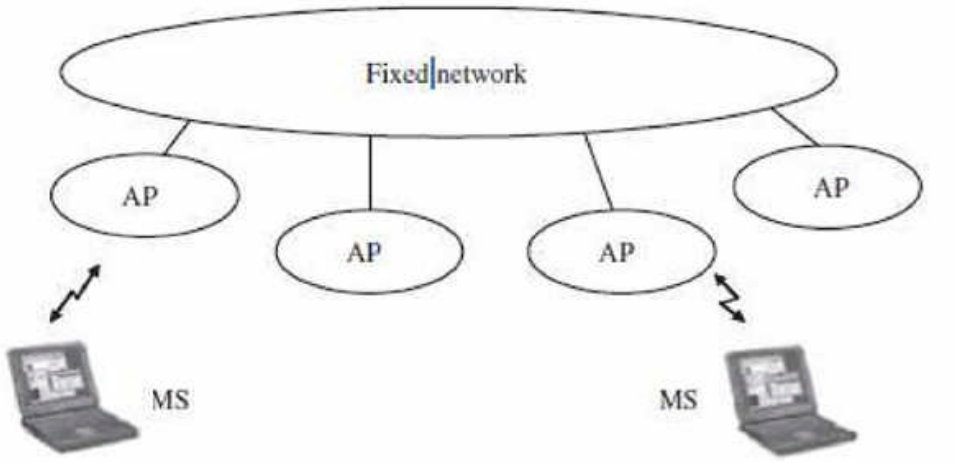
- Type 1 has a distributed MAC with QoS provisions, whereas type 2 has a centralized schedule MAC.
- Type 1 is based on Gaussian minimum shift keying (GMSK), whereas type 2 is based on OFDM.
- HiperLAN/2 automatically performs handoff to the nearest access point. The access point is basically a radio BS that covers an area of about 30 to 150 meters, depending on the environment. MANETs can also be created easily.

The goals of HiperLAN are as follows:

- QoS (to build multiservice network)
- Strong security
- Handoff when moving between local area and wide areas
- Increased throughput
- Ease of use, deployment, and maintenance
- Affordability
- Scalability

One of the primary features of HiperLAN/2 is its high speed transmission rates (up to 54 Mbps). It uses a modulation method called OFDM to transmit analog signals. It is connection oriented, and traffic is transmitted on bidirectional links for unicast traffic and unidirectional links toward the MSs for multicast and broadcast traff

This connection oriented approach makes support for QoS easy, which in turn depends on how the HiperLAN/2 network incorporates with the fixed network using Ethernet, ATM, or IP.

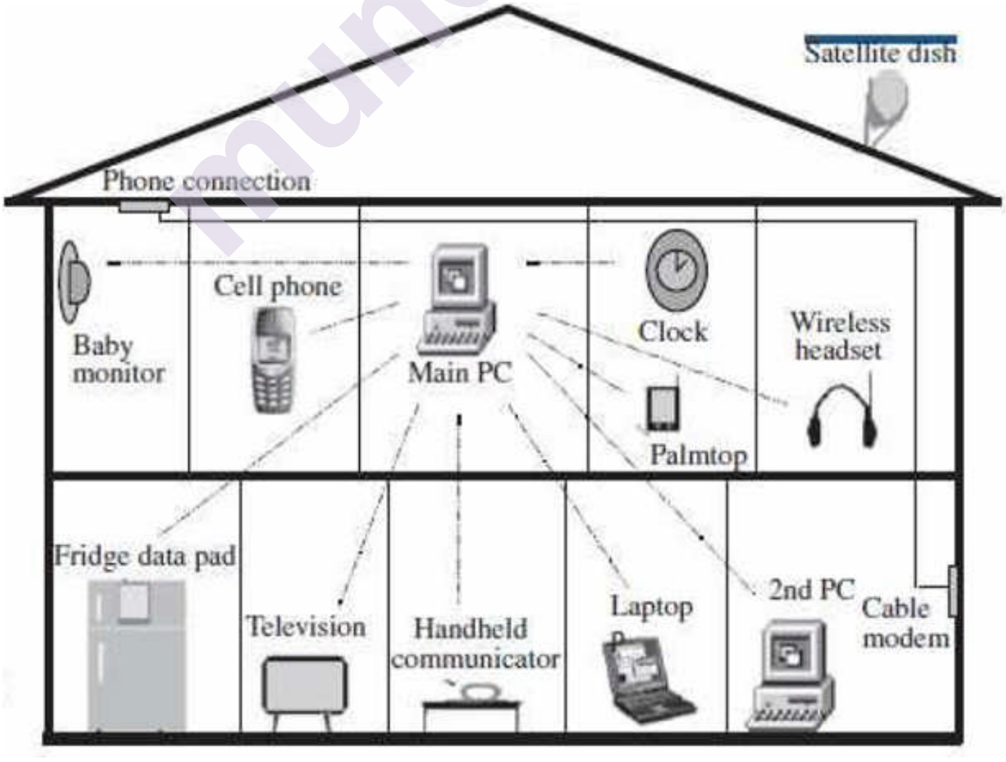


The HiperLAN/2 architecture shown in the figure allows for interoperation with virtually any type of fixed network, making the technology both network and application independent.

Hiper LAN/2 networks can be deployed at "hot spot" areas such as airports and hotels, as an easy way of offering remote access and internet services.

2. Home RF Technology

- A typical home needs a network inside the house for access to a public network telephone and internet, entertainment networks (cable television, digital audio and video with the IEEE 1394), transfer and sharing of data and resources (printer, internet connection), and home control and automation.
- The device should be able to self-configure and maintain connectivity with the network. The devices need to be plug and play enabled so that they are available to all other clients on the network as soon as they are switched on, which requires automatic device discovery and identification in the system.
- Home networking technology should also be able to accommodate any and all lookup services, such as Jini. Home RF products allow you to simultaneously share a single internet connection with all of your computers - without the hassle of new wires, cables or jacks.
- Home RF visualizes a home network as shown in the figure:



- A network consists of resource providers, which are gateways to different resources like phone lines, cable modem, satellite dish, and so on, and the devices connected to them such as cordless phone, printers and fileservers, and TV.
- The goal of Home RF is to integrate all of these into a single network suitable for all applications and to remove all wires and utilize RF links in the network suitable for all applications.
- This includes sharing PC, printer, fileserver, phone, internet connection, and so on, enabling multiplayer gaming using different PCs and consoles inside the home, and providing complete control on all devices from a single mobile controller.
- With Home RF, a cordless phone can connect to PSTN but also connect through a PC for enhanced services. Home RF makes an assumption that simultaneous support for both voice and data is needed.

Advantages of Home RF

- In Home RF all devices can share the same connection, for voice or data at the same time.
- Home RF provides the foundation for a broad range of interoperable consumer devices for wireless digital communication between PCs and consumer electronic devices anywhere in and around the home.
- The working group includes Compaq computer corp. Ericson enterprise network, IBM Intel corp., Motorola corp. and other.
- A specification for wireless communication in the home called the shared wireless access protocol (SWAP) has been developed.

3. IEEE 802.11 Standard

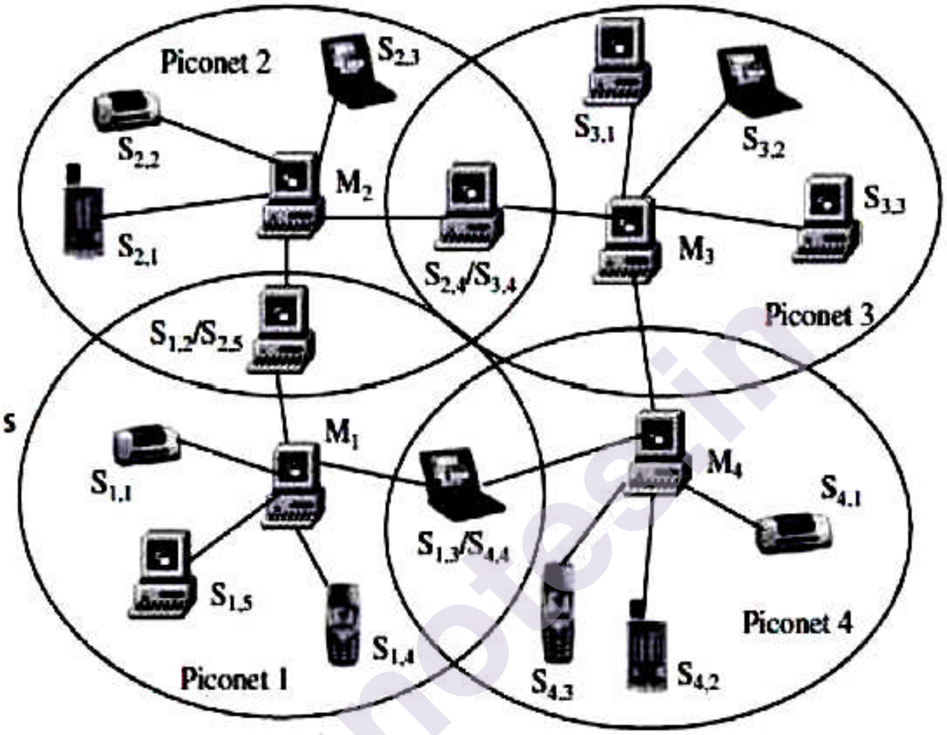
IEEE 802.11 is a set of standards for the wireless area network (WLAN), which was implemented in 1997 and was used in the industrial, scientific, and medical (ISM) band. IEEE 802.11 was quickly implemented throughout a wide region, but under its standards the network occasionally receives interference from devices such as cordless phones and microwave ovens. The aim of IEEE 802.11 is to provide wireless network connection for fixed, portable, and moving stations within ten to hundreds of meters with one medium access control (MAC) and several physical layer specifications. This was later called 802.11a. The major protocols include IEEE 802.11n; their most significant differences lie in the specification of the PHY layer.

4. Bluetooth

Bluetooth is one of the major wireless technologies developed to achieve WPAN (wireless personal area network). It is used to connect devices of different functions such as telephones, computers (laptop or desktop), notebooks, cameras, printers, and so on.

Architecture of Bluetooth

- Bluetooth devices can interact with other Bluetooth devices in several ways in the figure. In the simplest scheme, one of the devices acts as the master and (up to) seven other slaves.
- A network with a master and one or more slaves associated with it is known as a piconet. A single channel (and bandwidth) is shared among all devices in the piconet.



- Each of the active slaves has an assigned 3-bit active member address. many other slaves can remain synchronized to the master though remaining inactive slaves, referred to as parked nodes.
- The master regulates channel access for all active nodes and parked nodes. Of two piconets are close to each other, they have overlapping coverage areas.
- This scenario, in which nodes of two piconets intermingle, is called a scatternet. Slaves in one piconet can participate in another piconet as either a master or slave through time division multiplexing.
- In a scatternet, the two (or more) piconets are not synchronized in either time or frequency. Each of the piconets operates in its own frequency hopping channel, and any devices in multiple piconets participate at the appropriate time via time division multiplexing.
- The Bluetooth baseband technology supports two link types. Synchronous connection oriented (SCO) types, used primarily for voice, and asynchronous connectionless (ACL) type, essentially for packet data.

6.6 INFRARED VS RADIO TRANSMISSION

Infrared Transmission

- Infrared technology uses diffuse light reflected at walls, furniture etc. or a directed light if a line of sight (LOS) exists between sender and receiver.
- Infrared light is the part of the electromagnetic spectrum, and is an electromagnetic form of radiation. It comes from the heat and thermal radiation, and it is not visible to the naked eyes.
- In infrared transmission, senders can be simple light emitting diodes (LEDs) or laser diodes. Photodiodes act as receivers.
- Infrared is used in wireless technology devices or systems that convey data through infrared radiation. Infrared is electromagnetic energy at a wave length or wave lengths somewhat longer than those of red light.
- Infrared wireless is used for medium and short range communications and control. Infrared technology is used in instruction detectors; robot control system, medium range line of sight laser communication, cordless microphone, headsets, modems, and other peripheral devices.
- Infrared radiation is used in scientific, industrial, and medical application. Night vision devices using active near infrared illumination allow people and animals to be observed without the observer being detected.
- Infrared transmission technology refers to energy in the region of the electromagnetic radiation spectrum at wavelength longer than those of visible light but shorter than those of radio waves.
- Infrared technology allows computing devices to communicate via short range wireless signals. With infrared transmission, computers can transfer files and other digital data bidirectional.

Advantages of infrared

- The main advantage of infrared technology is its simple and extremely cheap senders and receivers which are integrated into nearly all mobile devices available today.
- No licenses are required for infrared and shielding is very simple.
- PDAs, laptops, notebooks, mobile phones etc. have an infrared data association (IrDA) interface.
- Electrical devices cannot interfere with infrared transmission.

Disadvantages of Infrared

- Disadvantages of infrared transmission are its low bandwidth compared to other LAN technologies.

- Limited transfer rates to 115 Kbit/s and we know that even 4 Mbit/s is not a particular high data rate.
- Their main disadvantage is that infrared is quite easily shielded.
- Infrared transmission cannot penetrate walls or other obstacles.
- Typically, for good transmission quality and high data rates a LOS (Line of site), i.e. direct connection is needed.

Radio Transmission

- Almost all networks use radio waves for data transmission, e.g., GSM at 900, 1800, and 1900 MHz, DECT at 1880 MHz etc. Radio transmission technologies can be used to set up ad-hoc connections for work groups, to connect, e.g., a desktop with a printer without a wire, or to support mobility within a small area.
- The two main types of radio transmission are AM (Amplitude Modulation) and (FM) Frequency Modulation.
- FM minimizes noise and provides greater reliability. Both AM and FM process sounds in patterns that are always varying of electrical signals.
- In an AM transmission the carrier wave has a constant frequency, but the strength of the wave varies. The FM transmission is just the opposite; the wave has constant amplitude but a varying frequency.
- Usually the radio transmission is used in the transmission of sounds and pictures. Such as, voice, music and television.
- The images and sounds are converted into electrical signals by a microphone or video camera. The signals are amplified, and transmitted. If the carrier is amplified it can be applied to an antenna.
- The antenna converts the electrical signals into electromagnetic waves and sends them out or they can be received. The antenna consists commonly of a wire or set of wires.

Advantages of Radio Transmission

- Advantages of radio transmission include the long-term experiences made with radio transmission for wide area networks (e.g. microwave links) and mobile cellular phones.
- Radio transmission can cover larger areas and can penetrate (thinner) walls, plants, furniture etc.
- Additional coverage is gained by reflection.
- Radio typically does not need a LOS (Line of Site) if the frequencies are not too high.
- Higher transmission rates (e.g. 54 Mbit/s) than infrared (directed laser links, which offer data rates well above 100 Mbit/s).

Disadvantages of Radio Transmission

- Radio transmission can be interfered with other senders, or electrical devices can destroy data transmitted via radio.
- Bluetooth is simple than infrared.
- Radio is only permitted in certain frequency bands.
- Shielding is not so simple.
- Very limited ranges of license free bands are available worldwide and those that are available are not the same in all countries.
- A lot harmonization is going on due to market pressure.

Transmission techniques

MAC Protocol issues
Wireless PANs
The Bluetooth technology

❖❖❖❖

Module IV

7

WIRELESS SENSOR NETWORKS

Unit Structure

7.0 OBJECTIVES

This chapter would make you understand the following concepts

- Implementation and use of wireless sensor networks
- To understand placement of sensor nodes to promote productivity
- Implement and evaluate new ideas for solving wireless sensor network design issues
- Various applications of wireless sensor networks

7.1 NEED AND APPLICATION OF SENSOR NETWORKS

7.1.1 Introduction

A sensor is a device that measures a change in a physical or environmental condition

In other words, a sensor is a device that responds to any thing or reaction, for example, any environmental parameters such as light, heat, humidity, or pressure, and generates a signal that can be interpreted and measured to produce output and display to the users for information purposes.

The Sensor Network community oftenly states a sensor node as a small, wireless sensing device, which has the ability to respond to the action, then process the captured data and transmit the data over the wireless connection using radio link.

Mostly sensors are implemented for measuring the environmental parameters such as light, heat, humidity, pressure, temperature. But it also has the ability to measure other factors too such as the vibrations, electromagnetic fields to predict any natural calamities.

The information captured by these small units called the sensors can be transmitted using wireless links so as to reduce the configuration complexity and make the information process simple and dynamic.

Once we have an idea about what are sensors, now let's look into what is a sensor network.

Sensor Network

A Sensor Network is an ad hoc wireless network which is made of a large number (hundreds or thousands) of sensor nodes, which are positioned randomly.
A sensor network communications consist of a protocol stack model.

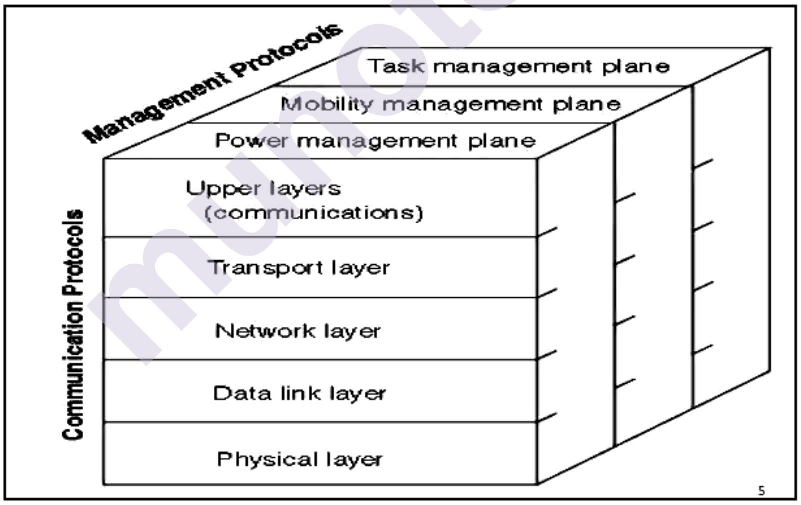


Figure: Protocol stack of Sensor Network

Protocol stack model is broadly classified into:

1. Communication protocol
2. Management protocol

1. **Communication Protocol**
   It consists of a physical layer, data link layer, network layer, transport layer and application layer

2. **Management Protocol**

It consists of power management, mobility management and task management planes.

Wireless sensor networks (WSNs)

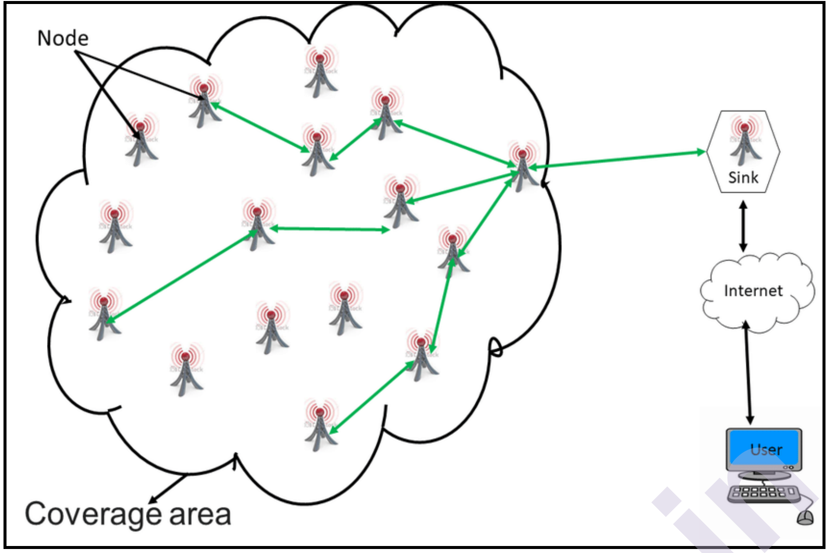


Figure: Wireless Sensor Network

Wireless Sensor Networks (WSNs) can be identified as a self-configured and infrastructure-less wireless networks which is formed by hundreds or thousands of sensor nodes which monitors physical or environmental conditions, such as sound, pressure, temperature, vibration, motion or pollutants and passes their data through the network to a base station (sink node) where the data is basically collected, observed and analysed.

A base or sink station acts like an interface (mediator) between the network and the user.

Every sensor node is equipped with computing and sensing devices, power components and radio transceivers.

After the sensor nodes are configured, they self-organize a considerable network infrastructure with multi-hop communication.

Local Positioning algorithms and Global Positioning System (GPS) are used to obtain information of location and position of sensor nodes.

These networks are also known as Actuator Networks because they consist of actuators.

7.1.2 Application of Sensor Network

WSNs has been implemented in various application domains to solve problems and make a change in our lives in many different ways

Military applications:

WSNs have a vital role to play in military control, communication, command, computing, battlefield surveillance, intelligence, reconnaissance and targeting systems.

Area monitoring:

The sensor nodes are installed sparsely over an area and the phenomenon is monitored. The monitored parameters (humidity, temperature) are then sent to the base stations to take appropriate actions.

Transportation:

WSN helps in collecting traffic information to alert the drivers regarding the traffic and congestion problems in their path.

Health applications:

WSN also contributes in health applications in monitoring patients, conducting diagnosis, drug administration in hospitals, monitoring patient’s physiological data, and tracking & monitoring patients, doctors.

Environmental sensing:

WSN is popularly developed to cover many applications related to earth science research. This includes sensing earthquakes, glaciers, volcanoes, oceans, etc.

Some of the other areas are as follows:

- Air pollution monitoring
- Greenhouse monitoring
- Forest fires detection
- Landslide detection

Structural monitoring:

Sensors can be used to inspect the building structures, monitor the movement within buildings and durability of the infrastructure such as bridges, flyovers, tunnels etc. The result of the monitoring helps the civil architect to take preventive measures against any hazard

Industrial monitoring:

WSN has been developed for implementing machinery on condition-based maintenance (CBM) as it promotes cost savings and enables new features and increases productivity.

Agricultural sector:

Wireless sensor network measures the humidity of the soil and indicates the need of irritation of crops thus increasing the productivity. It also promotes irrigation automation thus enabling use of water efficiently and reducing wastage of water.

7.2 SENSOR NETWORKS DESIGN CONSIDERATION

WSN consists of a large number of sensor nodes that have less processing capability and limited power. The number of sensor nodes in the network are not constant i.e. there can be addition of new sensor nodes in the network or nodes may be deleted also.

While designing the sensor network, the factors that are to be considered are listed below.

Fault Tolerance:

There is a possibility that a node fails, thus changing the topology of the network. In such a case, the network must be made robust to adopt the changes so that the functioning of the topology is not disrupted and the network functions efficiently.

Lifetime:

It is assumed that WSN should work for at least 6 months to 1 year using a 3 V battery providing good performance and with good energy. The designer should keep in mind that it consumes less energy thus making the network to last for longer.

Scalability:

The WSN must be able to support additional nodes at any given point of time without interfering with the other's performance. Also some applications require more number of sensor nodes so the design should be made in such a way that it supports a large design of network.

Date Aggregation:

The sensor nodes are placed close to each other due to which similar data can be generated by the nodes which are next to each other. The data can be collected and duplicate data can be extracted at different levels.

Cost:

The cost of each sensor node is very expensive and as we all know the Sensor network is made up of a large number of sensor nodes and hence the cost will be a major concern. So, the design must be such that all the data is monitored by optimal usage of the number of nodes.

Environment:

The sensor nodes deployed in WSN must be survivable under all conditions as the environment may be demanding.

Heterogeneity Support:

The design of the sensor network must support different types of sensor nodes based on its functionality.

Autonomous Operations:

The WSN should be able to operate, organize , and reorganize on its own without human intervention.

Limited Memory and Processing Capability:

Depending on the functionality, each sensor node has restricted power, memory, and processing capabilities, so the design should be such that additional memory or power is not needed.

7.3 EMPIRICAL ENERGY CONSUMPTION

A challenge for the design of WSN is minimizing the energy consumption of Wireless Sensors.

The energy consumption in WSN involves three components:

1. Sensing Unit (Sensing transducer and A/D Converter)
2. Communication Unit (transmission and receiver radio)
3. Computing/Processing Unit.

If we want to conserve energy then we'll have to put some SNs to sleep mode.

- **Sensing Unit:**

○ The Sensing transducer captures the physical parameters of the environment.

○ It does physical signal sampling and then converts it into electrical signals.

○ Using this component, the energy consumption depends on the hardware, the application used and the sensing energy spent.

- An AD Converter for sensor consumes only 3.1 JLlW , in 31 pJ/8-bit sample at lVolt supply.

The standby power consumption at IV supply is 4lpW.

Transmission Energy:
Energy consumed to transmit information is given as follows
$E_{Tx}(k,d)=E_{Tx\text{-}elec}(k)+E_{Tx\text{-}amp}(k,d)=E_{elec}*k+\varepsilon*k*d^2$, where
$E_{Tx\text{-}elec}$ is the transmission electronics energy consumption, $E_{Tx\text{-}amp}$ is the transmit amplifier energy consumption. Their model assumes
$E_{Tx\text{-}elec}=E_{Rx\text{-}elec}=E_{elec}=50nJ/bit$, and $\varepsilon_{amp}=100pJ/bit/m^2$

Receiver Energy:
Energy consumed to receive k bits.

$E_{Rx(k)}=E_{Rx\text{-}elec}(k)=E_{elec}*k$

- **Commutation Unit:**

The computation unit consists of a microcontroller/ processor with memory which can monitor, control and operate the sensing, computing and communication unit.

The energy consumption of this unit is categorised into two parts:

- switching energy
- leakage energy

Switching energy is calculated as

$$E_{switch}=C_{total}V_{dd}^{2};$$

where Ctotal is the total capacitance switched by the computation and Vdd is the supply voltage.

When no computation is carried out, the energy consumed is called leakage energy. It can be calculated as:

$$E_{leakage,up} = (V_{dd}t)I_0 e^{\frac{V_{dd}}{nV_T}}$$

where VT is the thermal voltage, I0 are the parameters of the processor Sleeping: To save energy, sensors can be put to sleep-active cycles. When a sensor is put to sleep, thus saving energy.

7.4 SENSING AND COMMUNICATION RANGE

A wireless sensor network (WSN) consists of a large number of sensor nodes (SNs) The main objective of a SN is to monitor physical and environmental parameters. In a given area, the sensors need to be deployed so that the complete area sensing can be done, without leaving any area not monitored. The SNs can be distributed randomly or placed at a preferred location.

Every SN has its own sensing range and to sense the complete area, the neighboring SNs have to be installed close to each other and at maximum 2rs distance from each other.

If the Sensor Nodes are uniformly distributed with the node having density X, then the probability of having 'm' Sensor nodes within the area of S is Poisson distributed as

$$P(m)=\frac{(\lambda S)^m}{m!}e^{-\lambda S}$$

This is basically the probability that the monitored area is not covered by any Sensor node and therefore the probability pcovcr of the coverage by at least one SN is:

$$p_{cover}=1-P(0)=1-e^{-\lambda S}$$

It gives us information about the coverage of the area so that we understand how many more sensor nodes are needed to be deployed.

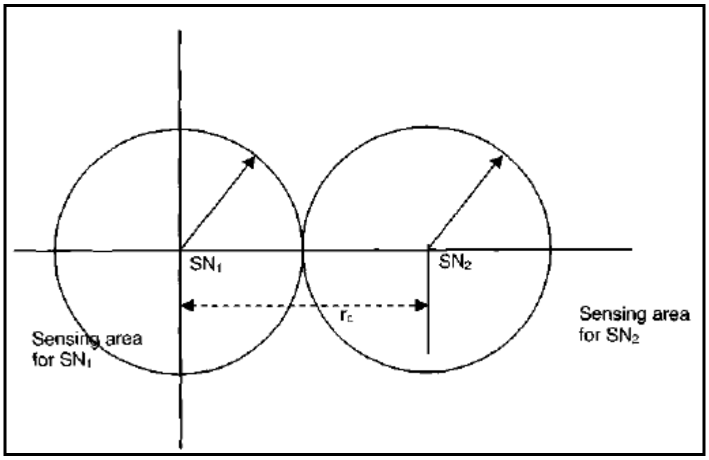
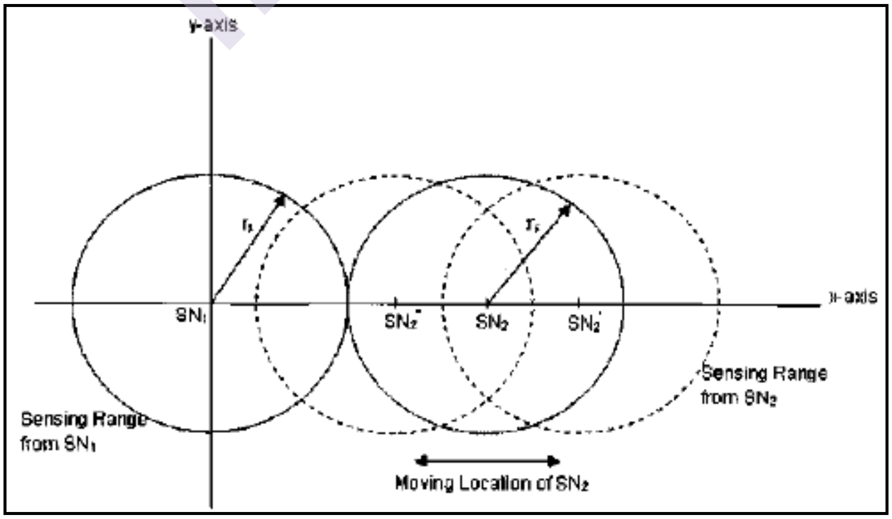


Figure: Sensing range of the sensor nodes

When there is at least one sensor node within the communication range then the transmission between neighboring Sensor Nodes is achievable

Information from one sensor node is not sufficient. We need a set of SNs to monitor the area and provide information so for this the major concern is how far should the sensor nodes be placed. The distance between two sensor nodes is estimated as 2rs from the sensing coverage point.

Let us consider SN1 and SN2, to establish a communication between these two SNs the minimum distance between them should be 2rs i.e. the communication coverage range should be at least twice the sensing distance



7.5 LOCALIZATION SCHEME

In Wireless Sensor Networks (WSNs), Localization is to discover the current location of the sensor nodes. Localization is calculated by the communication between localized and unlocalized sensor nodes for deciding their geometrical position. Location is nothing but distance and angle between nodes.

The many concepts used in localization are stated as follows.

- Lateration - Distance between nodes is measured.
- Angulation - Angle between nodes is measured
- Trilateration - It is the distance measurement from three nodes. Position of an unlocalized node is calculated by the intersection of three circles and the single point is the position.
- Multilateration - To determine the location, more than three nodes are needed
- Triangulation - Minimum two angles has to be measured of an unlocalized node from localized sensor nodes.

Localization schemes are classified as:

- anchor based or anchor free
- centralized or distributed
- GPS based or GPS free
- fine grained or coarse grained
- stationary or mobile sensor nodes
- range based or range free.

- **Anchor Based and Anchor Free**

In anchor-based mechanisms, the few node positions are known. Location of unlocalized nodes are determined by the position of the known nodes. Accuracy is highly dependent on the number of anchor nodes.

In Anchor-free mechanism, algorithms estimate relative positions of nodes instead of computing absolute node positions

- **Centralized and Distributed**

In a centralized based algorithm, one central point known as sink node collects all the information from the nodes. This sink node or base station calculates the position of nodes and forwards the information to others. It consumes less energy as well as the Computation cost is less. It promotes clustering.

In distributed based algorithms, individually the sensors calculate and estimate their positions and they directly communicate with the sink node or the base station. It does not promote clustering.

- **GPS Based and GPS Free**

  In GPS-based schemes, every node has a GPS receiver which makes it very costly. And also the localization accuracy is very high.

  GPS-free algorithms are less expensive as they do not have GPS, and they calculate the distance between the nodes relative to the local network

- **Coarse Grained and Fine Grained**

  For coarse-grained localization schemes, the result is achieved using received signal strength.

  For Fine-grained localization schemes, the result is achieved using the received signal strength

- **Stationary and Mobile Sensor Nodes**

  Localization algorithms also depend on the field of sensor nodes on which they are deployed. Some nodes are fixed at one place i.e. they are static in nature because many applications prefer to use static nodes. Because of which, most of the localization algorithms are designed for static nodes. There are few applications that use mobile sensor nodes.

7.6 CLUSTERING OF SENSOR NETWORKS

Basically, clustering means grouping. So clustering of Sensor nodes indicates collecting the sensed data and limits the transmission of sensed data within the cluster so as to reduce traffic and congestion in network

For clustering the sensor nodes, we first need to discover the neighbors by sending the Beacon signals and the cluster head (CH) is also selected.

The major concern here is how to group neighboring Sensor nodes and how many clusters need to be formed for optimized performance.

One of the approaches is to partition the Wireless Sensor Network into clusters in such a manner that all members of the clusters are connected to the Cluster Head (CH) directly.

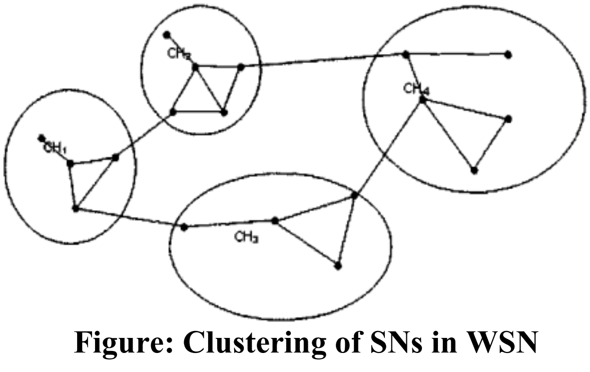


Figure: Clustering of SNs in WSN

The above figure shows randomly deployed SNs. The Sensor nodes in a cluster can transmit data to CH directly thus reducing the energy consumption.

The CHs also transmits information among themselves

The energy consumed in any wireless transmission is proportional to the square of the distance between the Sensor nodes and the Cluster head.

It can be convinient to partition SNs in a WSN into a d-cluster. Each SN within a cluster is expected to maintain a list of all members of a cluster.

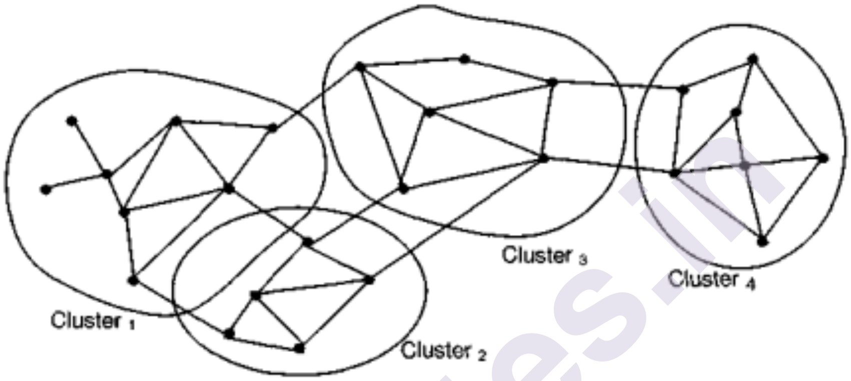


Figure: Clustering with d clusters

It is unrealistic to assume that hundreds or thousands of Sensor node will have the information about the whole WSN connectivity

How to select the CH of a cluster?

A simplest technique is by selecting the largest weight in the cluster.

Another technique is to use the Sensor node with the highest degree i.e. a node having the largest number of neighbors in the cluster. We need to select the CH calculatively because the data collected by the cluster members is trusted with the transmission of that data. The CH does most of the work so the CH may run out of energy. In such cases, a dynamic change in CH must be implemented.

The CH may allot different time slots to the cluster member for data transmission so as to avoid collision of data. So it is preferable for each SN to use one specific channel. Sufficient number of SNs are required to be deployed so as to monitor information of every corner of the area.

7.7 ROUTING LAYER

The question arises that how the nodes will communicate among themselves The routing protocols define how nodes will communicate with each other so as to share information throughout the network. The routing protocols in WSN can be classified in the following approach:

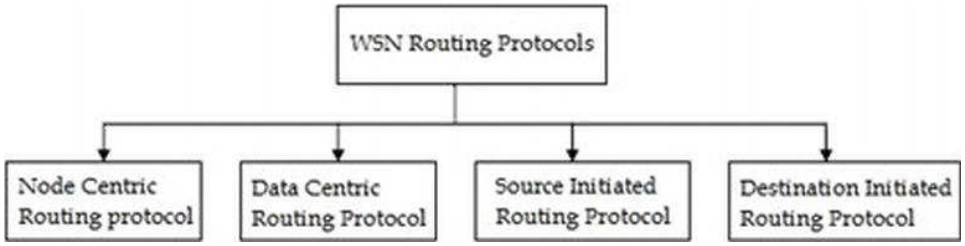

WSN Routing Protocols
Node Centric Routing protocol
Data Centric Routing Protocol
Source Initiated Routing Protocol
Destination Initiated Routing Protocol

Figure: WSN Routing Protocol approaches

1. Node centric approach

In this type of approach, every destination node is identified as a numeric identifier. Low energy adaptive clustering hierarchy (LEACH) is one of the protocols that implements a node centric approach .

- **Low energy adaptive clustering hierarchy (LEACH)**

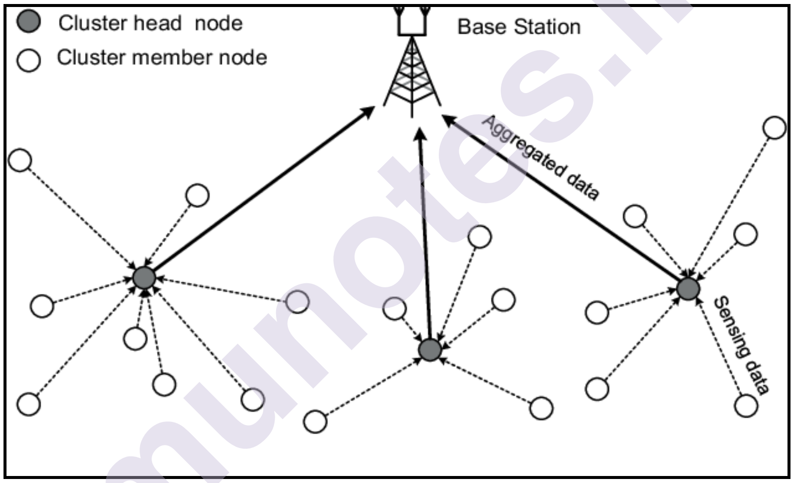


Figure: Low energy adaptive clustering hierarchy (LEACH)

Low energy adaptive clustering hierarchy (LEACH) routing protocol first organizes the cluster sensor nodes so that the energy is equally divided among all the sensor nodes in the network thus increasing the lifetime of the nodes. Using the LEACH protocol, we can form clusters such that a cluster head(CH) is selected and the remaining are sensor nodes before the communication begins. The Cluster head (CH) is assumed as a routing node for all the other member nodes in the specific cluster.

In LEACH, cluster head (CH) is either selected randomly from the cluster or some sensor nodes can volunteer themselves as a cluster head and inform the other nodes

2. **Data-centric approach**

In some wireless sensor networks, the transmission of sensed data is more important than how the data was collected collecting data from the nodes.

In a data centric routing approach, the sink node instructs the nodes to collect some specific characteristics.

Two of the protocol which follow this approach is as follows:

- **Sensor protocols for information via negotiation (SPIN)**

This protocol implements three messages namely

1. ADV
2. REQ
3. DATA

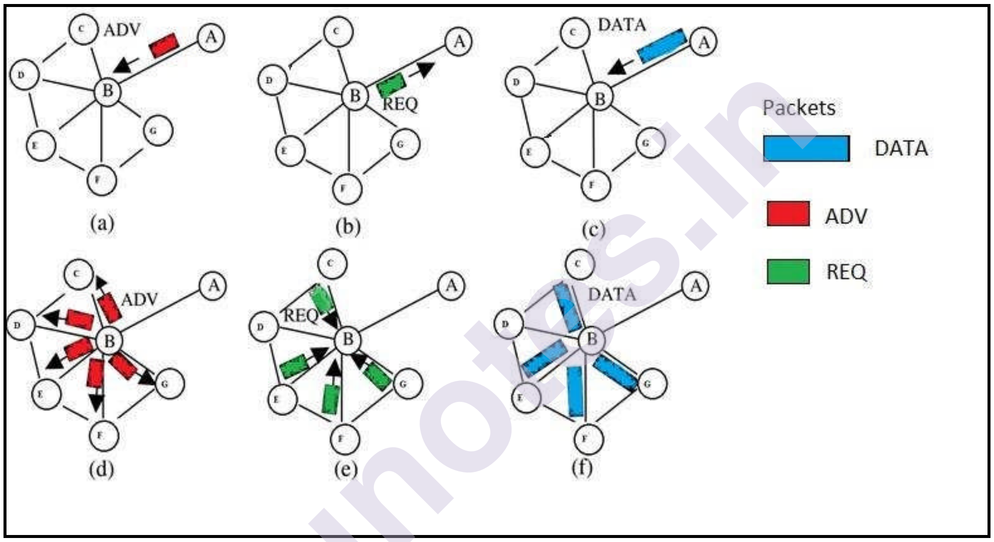


Figure: Working of SPIN Protocol

- The figure above shows the transmission of data using SPIN protocol.
- Firstly the node which has some information broadcasts an ADV packet to all its neighboring nodes.
- If any node is interested in knowing the data, then that node sends an REQ message to the advertising node.
- When the receiving node receives the REQ message from a specific node, it sends the actual DATA to that node who has shown an interest.
- Once the interested node receives the DATA. It further broadcast the ADV message to its neighboring nodes. In this way, the data is transmitted through the network.

- **Directed diffusion (DD)**

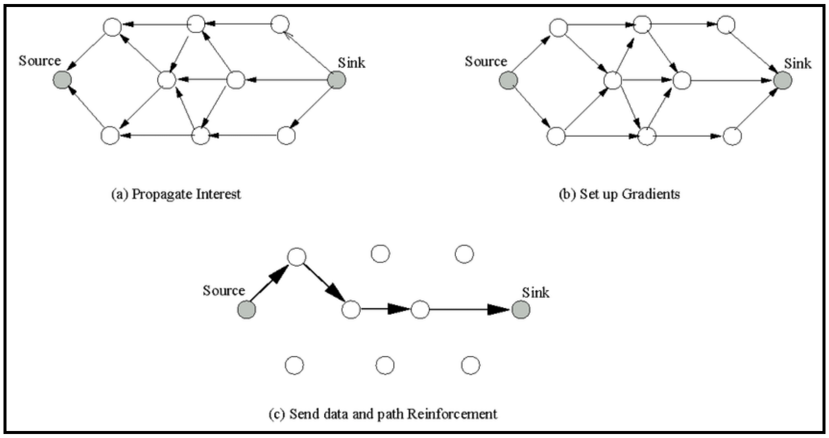


Figure: Directed diffusion

Directed diffusion is another data centric routing technique where importance is given to the data. It uses this approach for information gathering and transmission of data among the nodes. The Working is quite similar to that of the SPIN protocol.

As shown in the diagram, this approach has three steps:

- Propagate interest
- Set up the route
- Send data to the interested nodes

This routing protocol provides energy saving and efficiency thus increasing the lifetime of the network.

3. Source-initiated (Src-initiated)

In this approach, if the source node has data to share, it initiates a route from the source to the destination node.

Source-initiated can be implemented using SPIN Protocol

4. Destination-initiated (Dst-initiated)

Mostly, the route is generated by the source node but sometimes, the route generation is initiated by the destination node. If this has to be achieved then there are protocols needed to set up route generation.

Directed Diffusion (DD) & LEACH are the two protocols that implement the Destination-initiated approach.

7.8 SENSOR NETWORKS IN CONTROLLED ENVIRONMENT & ACTUATORS

Sensor networks in controlled environments and actuators is a group of sensors that are deployed at certain areas and gather information about their environment and actuators, such as motors or servos that interact with them. All elements communicate wirelessly in a human-controlled or autonomous manner.

It is also referred as wireless sensor and actor networks because there can be more than one actuator which will be involved on an actor point. An actor point can consist of a combination of multi-geared electric motors and servos that are arranged together to accomplish more complex functionalities. It supports automated measurement of environmental variables and can also have a control on various aspects of the environment directly through autonomous or controllable sensors and actors.

As they are built of multiple nodes whose involvement range differs from hundreds to thousands connected with more than one sensor with sensor hubs, individual actuators or actors.

Nowadays, It is mostly used where the measurements need to be accurate and precise such as telemedicine, monitoring industrial settings, entire population and scientific development.

Initially, Sensor networks in controlled environments and actuators were deployed by government and military agencies to monitor the persons, battlefields and other organizations and environments.

It is also contributing to the increasing trend of IoT (Internet of Things).

7.9 REGULARLY PLACED SENSOR

A simple strategy is to place the sensors in the form of a two dimensional grid as such a cross-point and such configuration may be very useful for uniform coverage if the area is easily accessible and the sensor can be placed anywhere. Such symmetric allocation of sensor nodes allows best possible regular coverage and clustering also is very easy to form with the neighboring SNs.

There are three samples of Sensor Networks namely rectangular, triangular and hexagonal clustering as shown in the above figure

The first figure is of the rectangular clustering of size 5x5 where a Sensor node is placed at each intersection of lines.

The rectangle, triangle, or hexagonal placement of the Sensor nodes indicate the minimum sensing area that needs to be covered by each sensor.

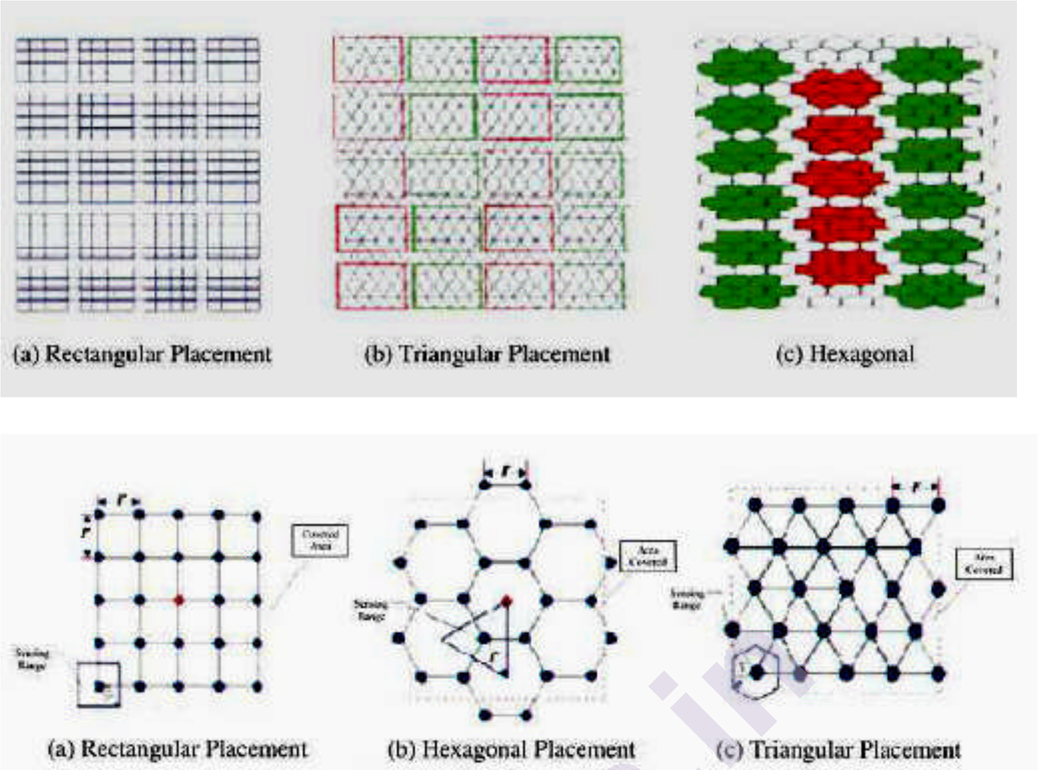
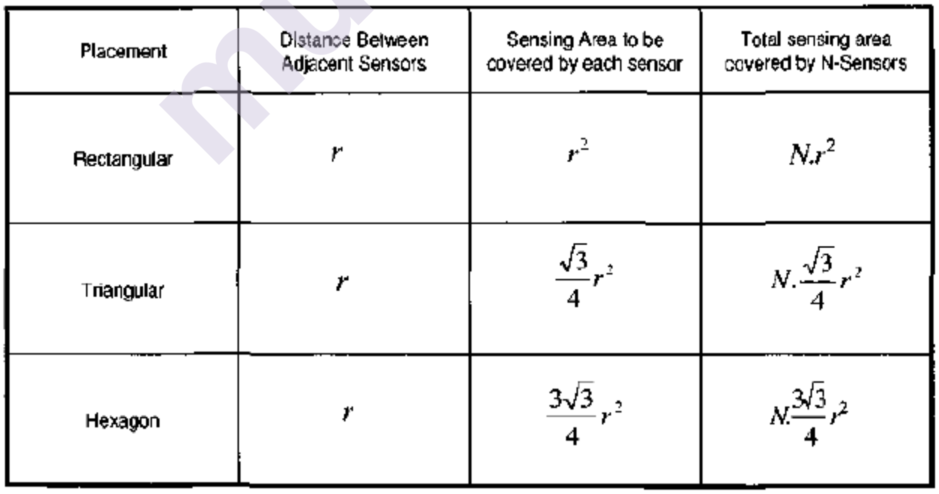


Figure: Two Dimensional grid format for sensor placement

Detailed representation of Sensor nodes in three different cluster samples, are shown in the figure above. The sensing area covered by rectangular allocation of sensor nodes is represented in a rectangular format, while sensing by triangular and hexagon placement is represented as triangular and hexagonal respectively.

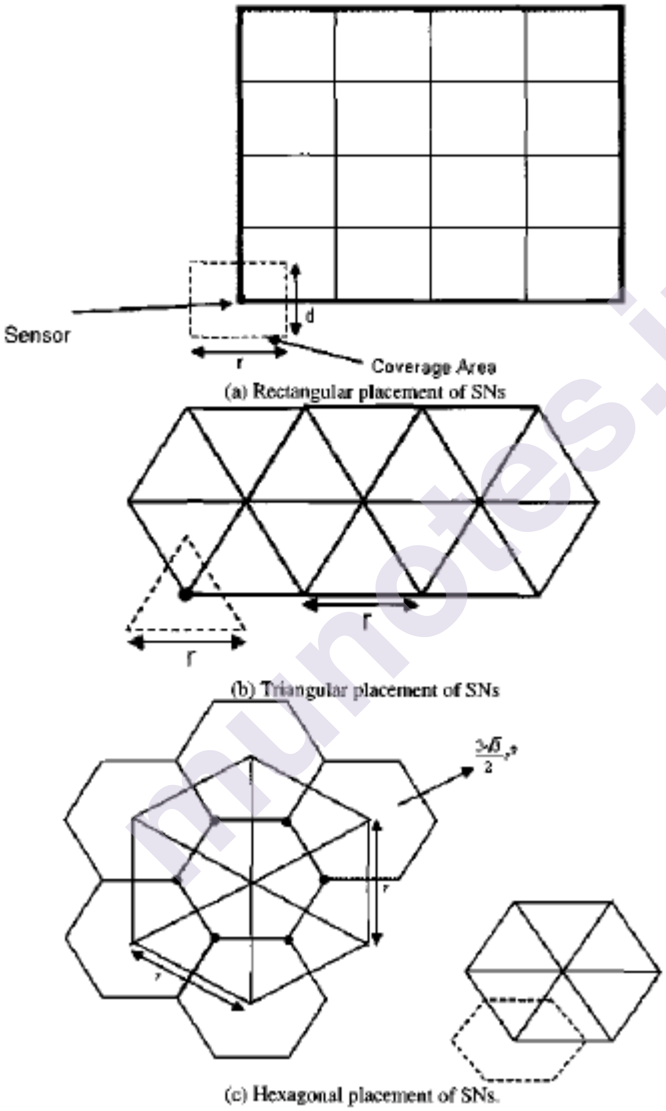


Placement	Distance Between Adjacent Sensors	Sensing Area to be covered by each sensor	Total sensing area covered by N-Sensors
Rectangular	r	r^2	$N.r^2$
Triangular	r	$\frac{\sqrt{3}}{4}r^2$	$N.\frac{\sqrt{3}}{4}r^2$
Hexagon	r	$\frac{3\sqrt{3}}{4}r^2$	$N.\frac{3\sqrt{3}}{4}r^2$

Figure: Placement of sensor and covered sensing range

The table above specifies the placement configuration of sensors and sensing range covered by the sensor nodes.

It may be noted that the radio transmission distance between adjacent SNs need to be such that the sensors can receive data from adjacent sensors using wireless radio. The three placements also promote clustering of the Sensor nodes and the size of each cluster can be fixed as per our requirements. If the sensing and radio transmission ranges are set to the minimum value, then all the SNs need to be active all the time to cover the area and function properly. If range is widened , then each sub-region will require to deploy more than one sensor node to monitor that range and some selected Sensor nodes can be put to sleep to save energy.



(a) Rectangular placement of SNs

(b) Triangular placement of SNs

(c) Hexagonal placement of SNs.

Figure: Detailed representation of SNs configuration

7.10 RFID AS PASSIVE SENSORS

In this era of technology, there is an evolution of many technologies that are providing support to the needs of the business in a cost-effective way.

One of the most popular technologies is RFID or radio-frequency identification, or RFID.

RFID sensors are categorised as active and passive.

Passive RFID systems can operate in either low frequency (LF) or high frequency (HF) or ultra-high frequency (UHF) radio bands

RFID tags are embedded in our day-to-day applications, such as employee badges, inventory control, retail security tags, pay terminals, and so.

Passive RFID Tags

Passive RFID are implemented using high-power readers that transmit low-frequency, high-power RF signals to battery-free tags. The circuit is activated by the antenna in the tag which in turn is activated by the amount of energy flowing to it.

Further, the reader receives a coded message by the tag at a different frequency. Passive RFID technology is used for determining theft and inventory tracking.

The range of Passive RFID is roughly around 1-5 meters from the Passive RFID reader, so a large number of readers will be required to track the location of an item.

How does Passive RFID work?

The RFID system is made up of three parts namely – an RFID reader or interrogator, an RFID antenna, and RFID tags. But passive RFID differs slightly which includes two main components namely the tag's antenna and the integrated circuit (IC) or microchip.

Initially, the Passive tags wait for a signal from an RFID reader. The reader then sends energy to an antenna which converts that energy into an Radio Frequency wave that is sent into the read zone. The RFID tag’s internal antenna draws in energy from the RF waves, once the tag is read within the read zone. The energy is then moved to the Integrated Circuit (IC) from the tag’s antenna and powers the chip which generates a signal back to the RF system. This is known as backscatter. The backscatter is nothing but a change in the electromagnetic or Radio Frequency wave, which is detected by the reader through the antenna which interprets the information.

The most basic structure is referred to as an RFID inlay.

There are many other types of passive RFID but they are categorized into two types-

1. hard tags
2. inlays tags.

Hard RFID tags -

Hard RFID tags are made of plastic, metal, ceramic and even rubber and are durable. They also come in all kinds of shapes and sizes and are designed for a unique function, or application.

Some of these tags will be supported within two or more groups.

- **High Temperature** – Some specific passive RFID tags are designed to resist extreme temperatures and accommodate different types of applications among themselves.
- **Rugged –** In outdoor environments, applications require a tag that can monitor dust, snow, ice, and debris in the environment.
- **Size** – When choosing an RFID tag, size is one of the major considerations. Applications tracking small or large items have specific size constraints when tracking small or large items.
- **Materials –** UHF metal-mount tags are used if an application requires tracking metal assets as these tags reduce the problems faced by UHF RFID around metal.
- **Embeddable –** An embeddable tags can be fit in small crevices and be covered to safeguard the RFID tag from harm.

Figure: A roll of Passive RFID inlays

Inlays tags are the RFID tags having high volumes, but of low cost.

These inlays are mainly grouped into three types:

1. **Dry Inlays** – An antenna and RFID microchip (IC) is attached to a material or a layer called a web. These inlays look like they have been glazed (coated) with no adhesive.
2. **Wet Inlays** – A RFID microchip and an antenna is attached to a substrate, usually PVT or PET, with an adhesive. These inlays can then be peeled off from their roll and stuck on an item.
3. **Paper Face Tags** – These tags are basically wet inlays with a poly face or a white paper. These tags are used for applications that need printed logos or numbers.

All Passive RFID tags do not operate at the same frequency.

Passive RFID tags operate at three different frequencies depending on parameters such as the attachment materials, read range, and the application options.

- **Low Frequency (LF)** : **125KHz to 134 KHz –**

A Long wavelength with a short read range of about 1 - 10 cms.

As it is not affected much by water or metal, this frequency is used with animal tracking.

- **High Frequency (HF): 13.56 MHz –**

Also called Near-Field Communication (NFC)

A medium wavelength with a medium read range of about 1 cm to 1m.

Data transmissions, DVD kiosks, access control applications, and passport security are done using this type of frequency.

- **Ultra High Frequency (UHF): 865 - 960 MHz –**

A high-energy wavelength beyond 1m having a long read range. Passive UHF tags can be read from an average distance of about 5 - 6 meters, but larger UHF tags can achieve up to 30+ meters of read range in ideal conditions. Race timing, file tracking, IT asset tracking, file tracking, and so on uses this type of frequency which needs more distance of read range.

7.11 UNIT END QUESTIONS

1. What is WSN? What are the applications of WSN?
2. State and explain the design consideration in SNs?
3. What is the sensing and communication range between two SNs?
4. What do you mean by localization?
5. How is clustering done in WSN? How is the Cluster Head selected?
6. Explain RFID as a passive sensor

❖❖❖❖