

WEB TECHNOLOGY

INTRODUCTION

DEBASISH CHATTERJEE,
ASSISTANT PROFESSOR
DEROZIO MEMORIAL COLLEGE
GUEST FACULTY , SNU





CHAPTER CONTENTS

01

Overview

02

Network of Networks

03

Intranet

04

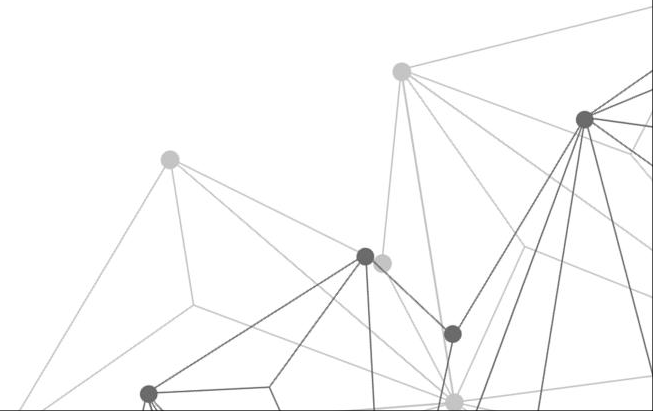
Extranet and Internet



01

Overview

This chapter helps you to understand the history and fundamental building blocks of Networks and WEB.



A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a web or a neural network structure.

What is Internet

Internet or Internetworking refers to a wide network through which computers are interconnected globally with one another and capable of sharing resources among themselves. This network is called internet which refers to millions of computers connected in a gigantic network communicated via TCP/IP Protocol.

Network of Network

An Internetwork may be defined as a network of computer communication networks every authorized member of which could communicate with every other authorized member (node) directly or indirectly.

It may consist of several Local, Metropolitan or Wide Area Networks interconnected via a LAN, MAN or a WAN oriented communication technology, depending upon the specific context of use.



Classification of Internetworks

There exist three classes of Internetworks for most of the practical and analytical purposes:

- **The Global Public Internetwork: The Internet**
- **The Wholly Owned / Private Internetworks: Intranets**
- **The Hybrid Internetwork-- private networks / internetworks connected through the Internet: Extranets**



INTERNET SERVICES AND ACCESSIBILITY

Today the Internet offers an extensive range of services many of which were not speculated when the Internet was first launched services.

- **Electronic mail:** A service that helps to send and receive messages and to attach files.
- **Electronic mailing lists:** Everyone subscribed to the list gets a message sent to the list.
- **USENET newsgroups:** Electronic bulletin board service
- **Real-time communication:** Chat, messengers, videoconferencing, white-boards, etc.
- **File Transfer Protocol (FTP):** A service that helps to send and receive files to and from a file server.
- **Telnet, ssh:** A remote login to other computers on the Internet.
- **Web World Wide Web (WWW):** Documents and files of various types which are connected using hypertext links to create a Web-like structure and are accessed through the Internet by addresses called Uniform Resource Locators (URLs).



USES OF THE INTERNET

- Telecommuting (working from home or anywhere)
- Online conferencing business,
- Advertising and online shopping
- News, jobs, softwares online courses, virtual classrooms,
- Coachings government services,
- Electronic publishing
- Entertainment (television, radio, videos, audio MP3s, etc.)
- Teaching and learning (course websites, conferencing, simulation, visualization, etc.)
- Scholarly research (searchable databases of journal articles; individual Web publishing, etc.)
- General information about a subject, especially which is not easily available from other sources Correspondence (email, chatting, etc.)



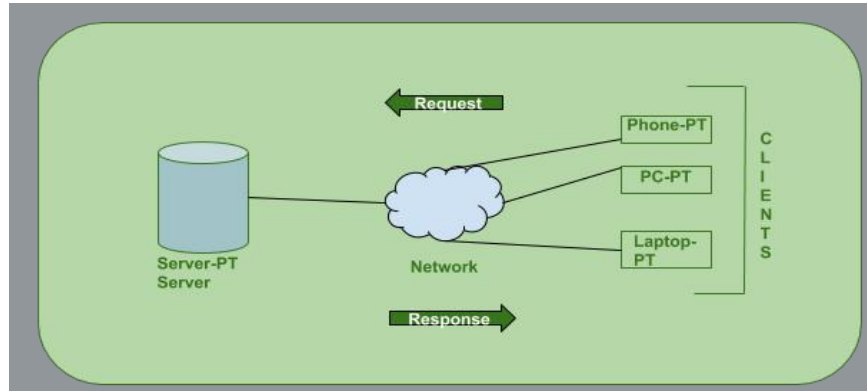
PROTOCOLS

Protocol is a set of rules or an agreement that specifies a common language that computers on a network use for communication with other computers.

- **Ethernet:** This is used to transfer information on a LAN. It specifies a number of wiring and signaling standards for the physical layer, two ways of network access (Media Access Control/Data Link Layer) and a common address format.
- **Internet Protocol (IP):** This protocol provides communicable global addresses of/to the computers. The computers identify each other by the IP addresses.
- **Transport Control Protocol (TCP):** This protocol guarantees reliable, proper delivery of data from the sender to the receiver. It breaks large messages, transports them reliably and reassembles them.
- **File Transfer Protocol (FTP):** This is used to connect two computers over the Internet so that the user of one computer can transfer files and perform file commands on the other computer. It exchanges files over any network that supports TCP/IP protocol.
- **Hypertext Transport Protocol (HTTP):** This protocol is used to retrieve Web pages from a Web server.
- **Simple Mail Transfer Protocol (SMTP):** This protocol is used for email transmissions.

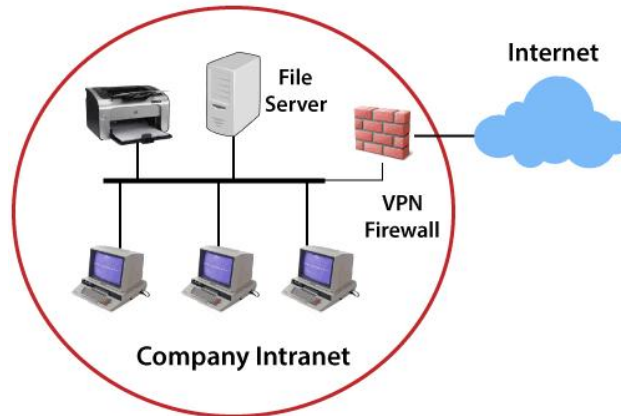
The Client/Server Model of the Web

- Most Internet services rely on the client/server model. The Internet user is the client and has the client software installed on his computer to access various Internet services.
- When a user wants to connect to a particular information tool, he uses his client software to connect to server programs, which provide the service or the information needed.
- The client/server model has become one of the central ideas of network computing, and is the basis of the TCP/IP protocol.



Intranet

- An intranet is a computer network for sharing information, easier communication, collaboration tools, operational systems, and other computing services **within an organization**, usually to the exclusion of access by outsiders.
- The term is used in contrast to public networks, such as the Internet, but uses most of the same technology based on the Internet protocol suite.





Intranet Use

- Intranets are being used to deliver tools, e.g. collaboration (to facilitate working in groups and teleconferencing) or sophisticated corporate directories, sales and customer relationship management tools, project management etc.,
- Intranets are also being used as corporate culture-change platforms.
- In large intranets, website traffic is often similar to public website traffic and can be better understood by using web metrics software to track overall activity.



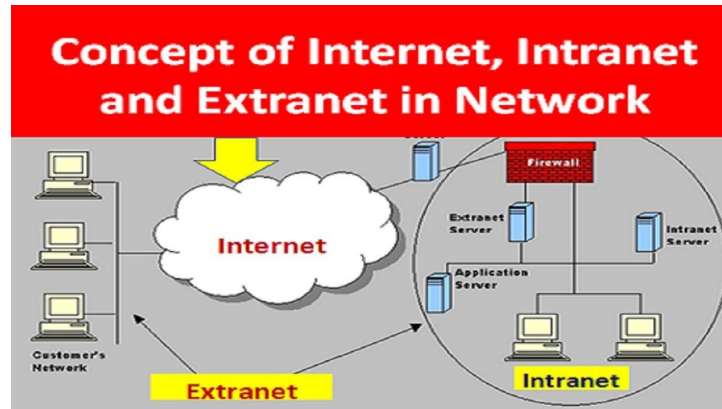
Extranet

An extranet is a private network that enterprises use to provide trusted third parties -- such as suppliers, vendors, partners, customers and other businesses -- secure, controlled access to business information or operations.

- Extranets, which take the form of external-facing websites or platforms, can sometimes be viewed as part of or an extension of the organization's intranet.
- Although information on an extranet is accessible to users outside the company, access is tightly controlled and only awarded to authorized users.

Some use cases for extranets

- Exchanging large volumes of data using electronic data interchange;
- Sharing product catalogs exclusively with wholesalers;
- Collaborating with other companies on joint development projects;
- Jointly developing and using training programs with other companies;
- Providing services to a group of other companies, such as an online banking application managed by one company on behalf of affiliated banks;
- Sharing news of common interest exclusively with partner companies.





CHAPTER CONTENTS

01

Domain and Sub domain

02

Address Resolution

03

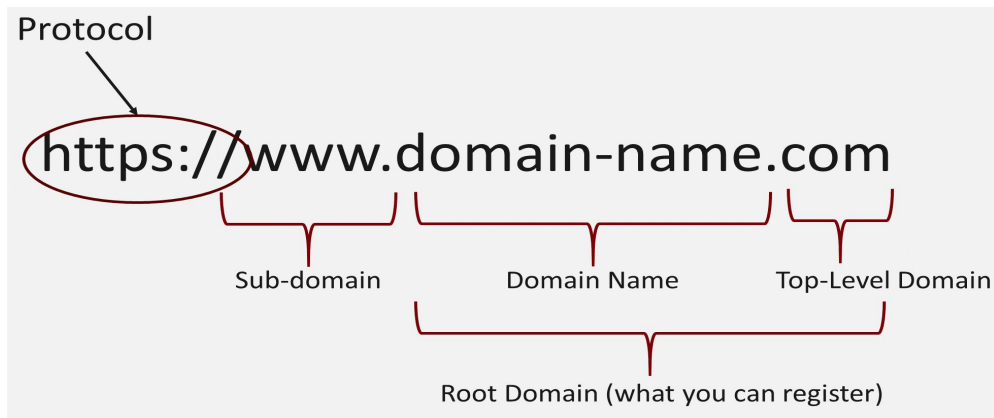
DNS

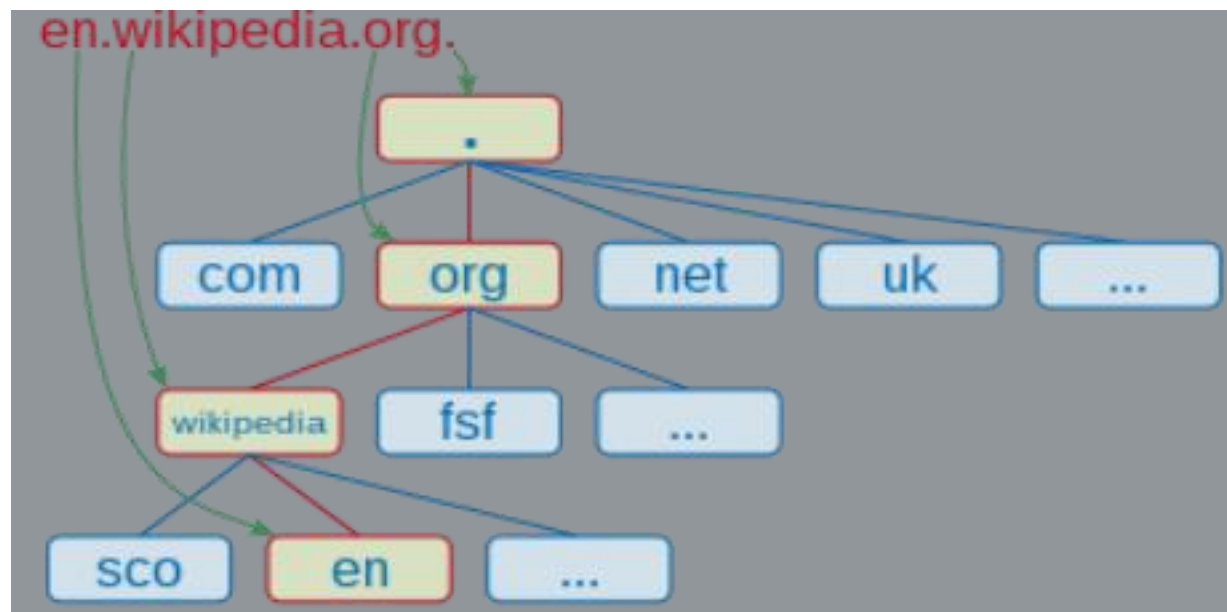
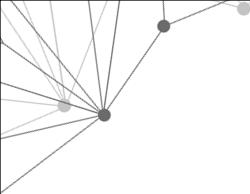
04

Telnet, FTP, HTTP

Domain

- A domain name is a string that identifies a realm of administrative autonomy, authority or control within the Internet.
- Domain names are used in various networking contexts and for application-specific naming and addressing purposes.
- Domain name identifies a network domain or an Internet Protocol (IP) resource, such as a personal computer used to access the Internet, or a server computer.







Subdomain

A subdomain is an additional part to your main domain name. Subdomains are created to organize and navigate to different sections of your website. You can create multiple subdomains or child domains on your main domain.

For example:

store.yourwebsite.com

In this example, '**store**' is the **subdomain**, 'yourwebsite' is the primary domain and '.com' is the top level domain (TLD).



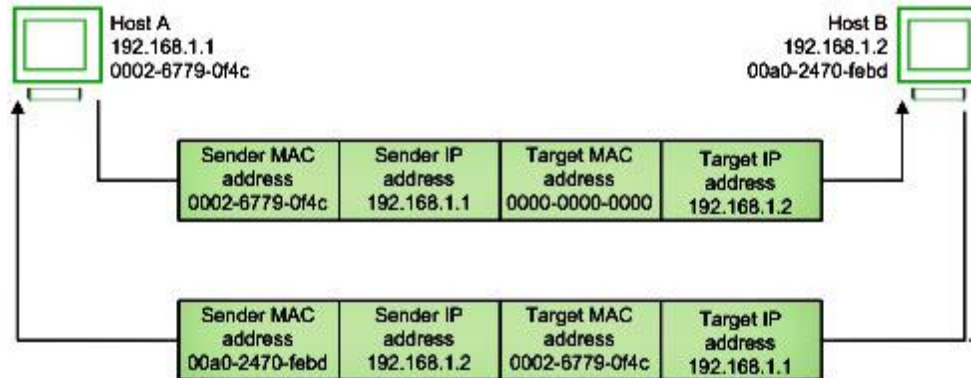
Use Cases of Subdomain

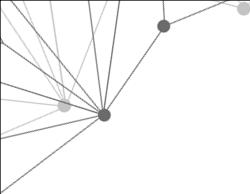
- For creating a testing or staging version of a website.
- Companies use subdomains for their mobile websites (m.yoursite.com), location-specific sites (uk.yoursite.com), and creating sub-sections of the website.
- You can install WordPress on your subdomain, and it will work as a separate installation from your main website.
- You can use a subdomain to serve a specific group of users on your site like 'guest.yourwebsite.com', 'user.yourwebsite.com'.
- Subdomains can be very useful in organizing your website content more efficiently. The right use of a subdomain does not affect your main website's SEO.

Address Resolution Protocol (ARP)

Most of the computer programs/applications use logical address (IP address) to send/receive messages, however, the actual communication happens over the physical address (MAC address). This is where ARP comes into the picture, its functionality is to translate IP address to physical addresses.

- ARP finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address.





The important terms associated with ARP are:

ARP Cache: After resolving the MAC address, the ARP sends it to the source where it is stored in a table for future reference. The subsequent communications can use the MAC address from the table.

ARP Cache Timeout: It indicates the time for which the MAC address in the ARP cache can reside.

ARP request: This is nothing but broadcasting a packet over the network to validate whether we came across the destination MAC address or not.

- The physical address of the sender.
- The IP address of the sender.
- The physical address of the receiver is FF:FF:FF:FF:FF:FF or 1's.
- The IP address of the receiver

ARP response/reply: It is the MAC address response that the source receives from the destination which aids in further communication of the data.

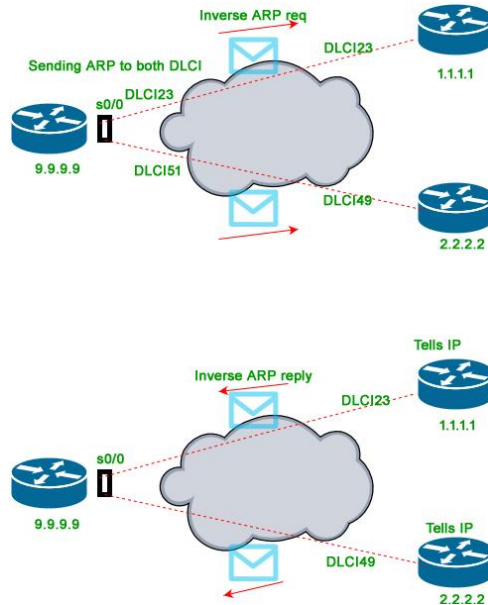
Reverse ARP

Reverse ARP is a networking protocol used by a client machine in a local area network to request its Internet Protocol address (IPv4) from the gateway-router's ARP table.



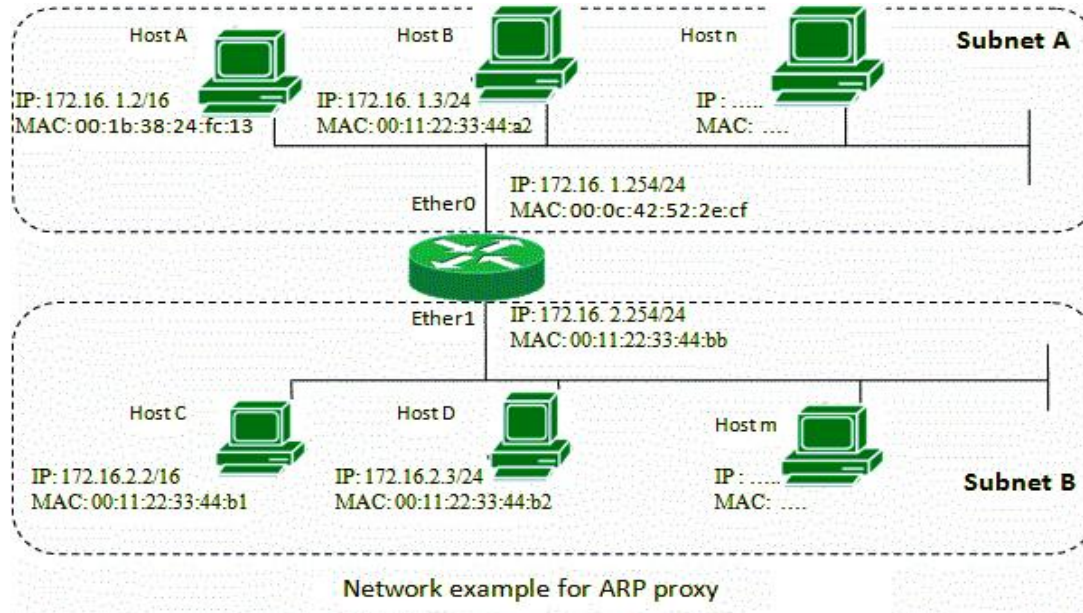
Inverse Address Resolution Protocol (InARP)

As the name suggests, InARP is just inverse of ARP. Reverse ARP has been replaced by BOOTP and later DHCP but Inverse ARP is solely used for device configuration.



Proxy ARP

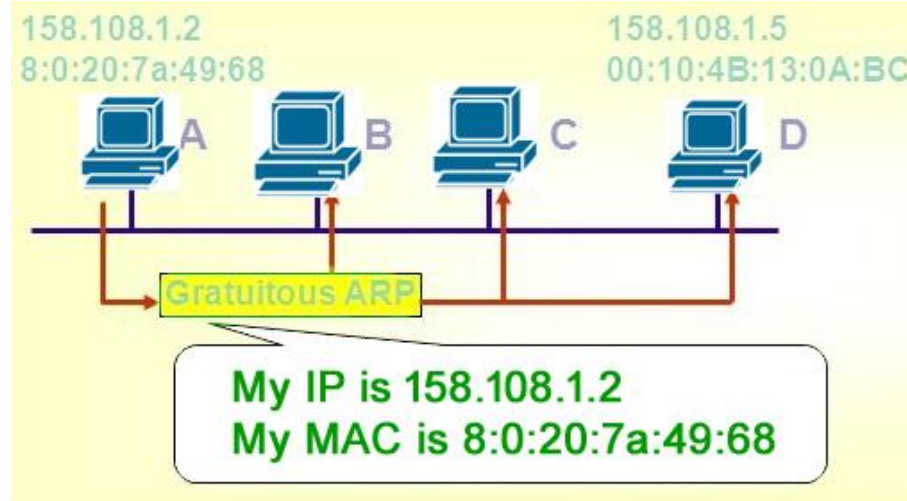
Proxy ARP was implemented to enable devices which are separated into network segments connected by a router in the same IP network or sub-network to resolve IP address to MAC addresses.



Gratuitous ARP

Gratuitous Address Resolution Protocol is used in advance network scenarios. It is something performed by computer while booting up. When the computer booted up (Network Interface Card is powered) for the first time, it automatically broadcast its MAC address to the entire network.

After Gratuitous ARP MAC address of the computer is known to every switch and allow DHCP servers to know where to send the IP address if requested.





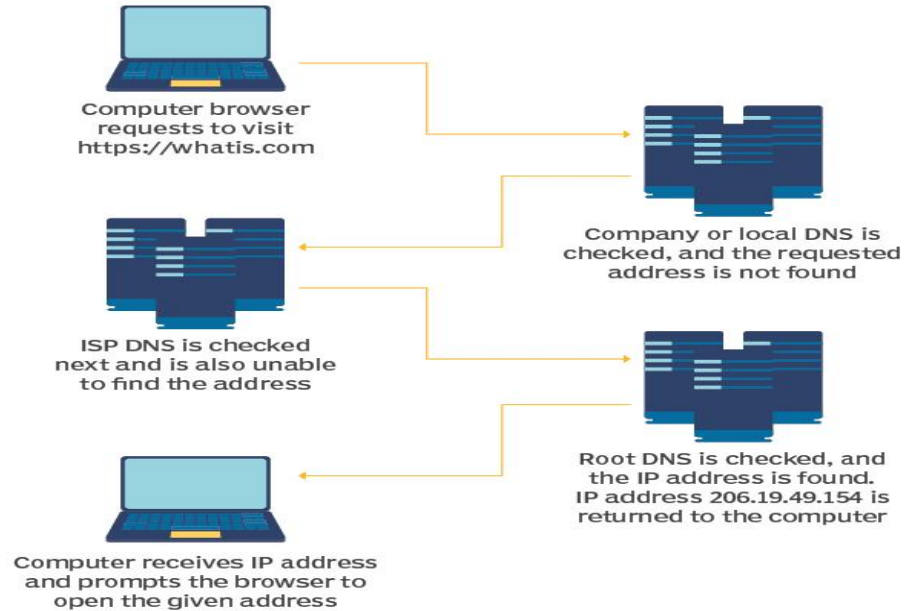
DNS

The domain name system (DNS) is a naming database in which internet domain names are located and translated into Internet Protocol (IP) addresses. The domain name system maps the name people use to locate a website to the IP address that a computer uses to locate that website.

How DNS works

DNS servers convert URLs and domain names into IP addresses that computers can understand and use. They translate what a user types into a browser into something the machine can use to find a webpage. This process of translation and lookup is called **DNS resolution.**

How DNS works



DNS servers talk to each other to answer a query from a client. Some DNS servers will have the necessary information cached and relay that back to the client so they can get online.



DNS structure

The domain name is usually contained in a URL. A domain name is made of multiple parts, called **labels**. The domain hierarchy is read from **right to left** with each section denoting a subdivision.

The TLD appears after the period in the domain name. Examples of top-level domains include **.com**, **.org** and **.edu**, but there are many others. Some may denote a country code or geographic location, such as **.us** for the **United States** or **.ca** for **Canada**.

- There can be up to 127 levels of subdomains.
- Each label can have up to 63 characters.
- The total domain character length can have up to 253 characters.
- Other rules include not starting or ending labels with hyphens and not having a fully numeric TLD name.



DNS Server Types

There are several server types involved in completing a DNS resolution. The following list describes the four name servers in the order a query passes through them.

- **Recursive server** : The recursive server takes DNS queries from an application, such as a web browser. It's the first resource the user accesses and either provides the answer to the query if it has it cached or accesses the next-level server if it doesn't.
- **Root name server** : The root name server is an index of all the servers that will have the information being queried.
- **TLD server**: The root server directs the query based on the top-level domain the .com, .edu or .org in the URL. This is a more specific part of the lookup.
- **Authoritative name server**: These servers know everything about a given domain and deal with the subdomain part of the domain name. They return the necessary record to the recursive server to send back to the client and cache it closer to the client for future lookups.



Types of DNS Queries

The following types of DNS queries are the main ones that take place at different points in the DNS resolution.

- **Recursive DNS queries** are those that take place between the recursive server and the client. Recursive queries end in either the answer or an error.
- **Iterative DNS queries** take place between the recursive resolver, which is a local DNS server, and the nonlocal name servers, like the root, TLD and authoritative name servers.
- **Nonrecursive queries** are those for which the recursive resolver already knows where to get the answer. If a recursive resolver has cached an IP address from a previous session and serves that address upon the next request, that is considered a nonrecursive query.



DNS Records

DNS records are the information a query seeks. Depending on the query, client or application, different information is required. Some records are required, such as the **A record**.

- **A record.** This stands for address and holds the IP address of a domain. A records only apply to IPv4 addresses.
- **NS record.** These name server records denote which authoritative server is responsible for having all the information about a given domain. Often, domains have both primary and backup name servers to increase reliability, and multiple NS records are used to direct queries to them.
- **TXT record.** TXT records enable administrators to enter text into DNS. TXT records are used to confirm domain ownership, secure email and counter email spam.
- **CNAME record.** Canonical name records are used instead of an A record when there is an alias. They are used to retry the query of the same IP address with two different domains.

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a web or a neural network structure.

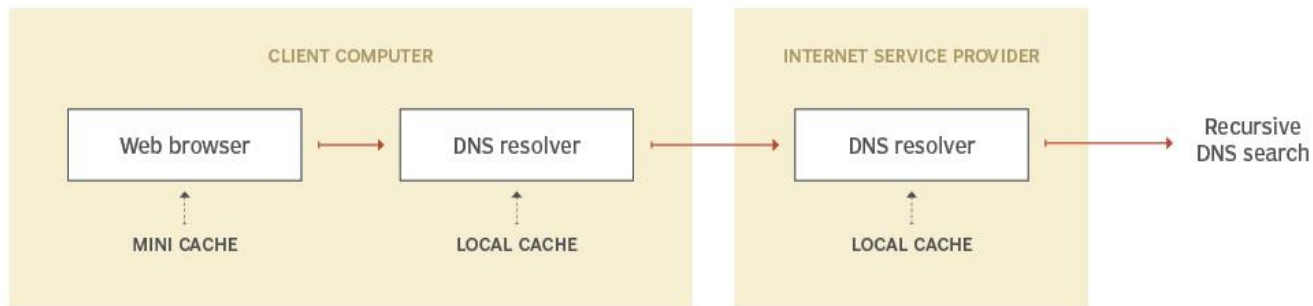
How does DNS increase web performance?

Servers can cache the A records, or IP addresses, they receive from DNS queries for a set amount of time. Caching promotes efficiency, enabling servers to respond quickly the next time a request for the same IP address comes in.

DNS data can be cached in a number of places. Some common ones include the following:

- **Browser.** The browser is the first cache that gets checked when a DNS request gets made, before the request leaves the machine for a local DNS resolver server.
- **Operating System (OS).** Many OSes have built-in DNS resolvers called stub resolvers that cache DNS data and handle queries before they are sent to an external server.
- **Recursive Resolver.** The answer to a DNS query can also be cached on the DNS recursive resolver. For example, if the resolver has A records but not NS records, the resolver can skip the root server and query the TLD server directly.

DNS caching flow



DNS queries look for the records in local caches, both on the DNS resolver within the operating system and on local applications, before queries are sent to external recursive servers.

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a web or a neural network structure.

Telnet

Telnet is a network protocol used to virtually access a computer and to provide a two-way, collaborative and text-based communication channel between two machines.

- It follows a user command Transmission Control Protocol/Internet Protocol (TCP/IP) networking protocol for creating remote sessions.
- Through Telnet, users can log on to a remote computer as a regular user with the privileges they are granted to the specific applications and data on that computer.
- While using telnet user are prompted to enter their username and password combination to access the remote computer, which enables the running of command lines as if logged in to the computer in person.



Uses of Telnet

- Telnet can be used for a variety of activities on a server, including editing files, running various programs and checking email.
- Users are also able to connect to any software that utilizes text-based, unencrypted protocols via Telnet, from web servers to ports.

Security

Telnet is not a secure protocol and is unencrypted. By monitoring a user's connection, anyone can access a person's username, password and other private information that is typed over the Telnet session in plaintext. With this information, access can be gained to the user's device.



FTP

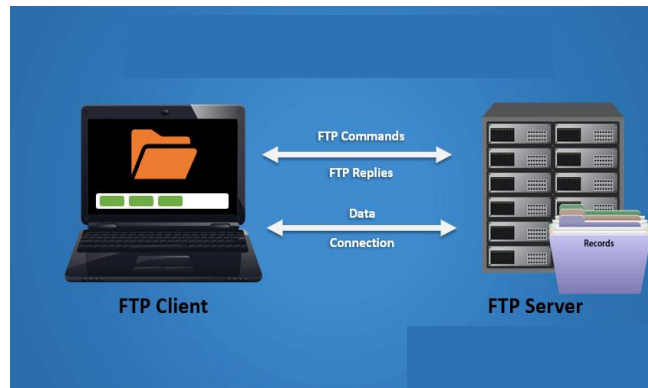
FTP (File Transfer Protocol) is a network protocol for transmitting files between computers over Transmission Control Protocol/Internet Protocol (TCP/IP) connections. Within the TCP/IP suite, **FTP is considered an application layer protocol.**

How does FTP work?

FTP is a client-server protocol that relies on two communications channels between the client and server: a command channel for controlling the conversation and a data channel for transmitting file content.

Here is how a typical FTP transfer works:

- A user typically needs to log on to the FTP server, although some servers make some or all of their content available without a login, a model known as **anonymous FTP**.
- The client initiates a conversation with the server when the user requests to download a file.
- Using FTP, a client can upload, download, delete, rename, move and copy files on a server.





FTP is used for file transfers for the following purposes :

- **Backup.** FTP can be used by backup services or individual users to backup data from one location to a secured backup server running FTP services.
- **Replication.** Similar to backup, replication involves duplication of data from one system to another but takes a more comprehensive approach to provide higher availability and resilience. FTP can also be used to facilitate this.
- **Access and Data loading.** FTP is also commonly used to access shared web hosting and cloud services as a mechanism to load data onto a remote system.

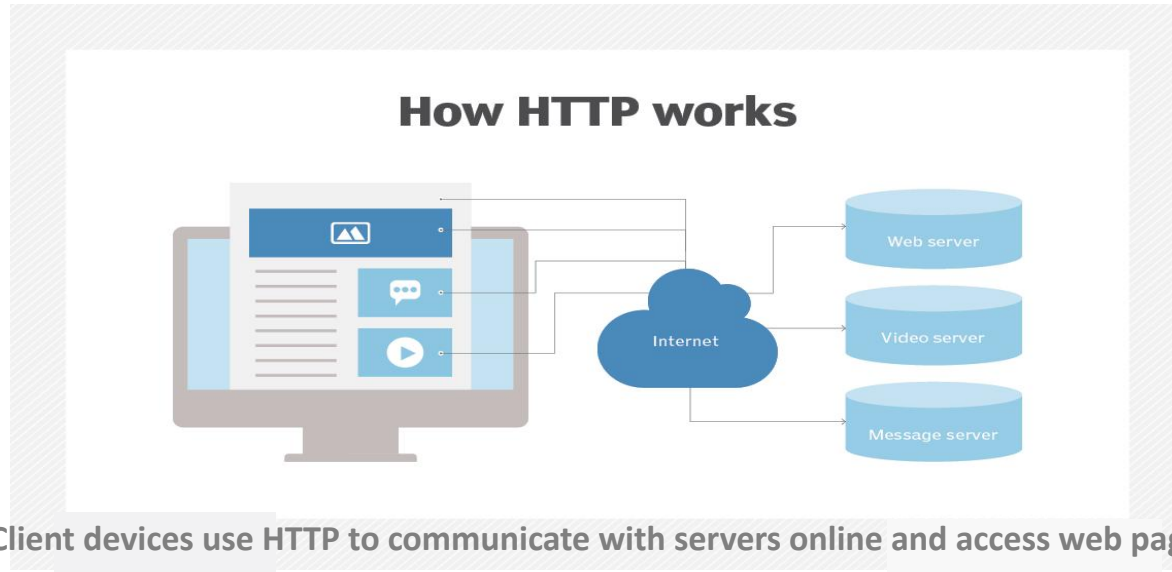


FTP Types

- **Anonymous FTP.** This is the most basic form of FTP. It provides support for data transfers without encrypting data or using a username and password. It's most commonly used for download of material that is allowed for unrestricted distribution. It works on port 21
- **Password-protected FTP.** This is also a basic FTP service, but it requires the use of a username and password, though the service might not be encrypted or secure. It also works on port 21.
- **FTP Secure (FTPS).** Sometimes referred to as FTP Secure Sockets Layer (FTP-SSL), this approach enables implicit Transport Layer Security (TLS) as soon as an FTP connection is established. It typically defaults to using port 990.
- **FTP over explicit SSL/TLS (FTPES).** This approach enables explicit TLS support by upgrading an FTP connection over port 21 to an encrypted connection. This is a commonly used approach by web and file sharing services to enable secure file transfers.
- **Secure FTP (SFTP).** This is technically not an FTP protocol, but it functions similarly. Rather, SFTP is a subset of the Secure Shell (SSH) protocol that runs over port 22. SSH is commonly used by systems administrators to remotely and securely access systems and applications, and SFTP provides a mechanism within SSH for secure file transfer.

HTTP (Hypertext Transfer Protocol)

HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files -- such as text, images, sound, video and other multimedia files -- over the web. HTTP is an application protocol that runs on top of the TCP/IP suite of protocols.





HTTP vs. HTTPS

- HTTPS is the use of Secure Sockets Layer (SSL) or Transport Layer Security (TLS) as a sublayer under regular HTTP application layering.
- HTTPS encrypts and decrypts user HTTP page requests as well as the pages that are returned by the web server.
- It also protects against eavesdropping and man-in-the-middle (MitM) attacks.
- Migrating from HTTP to HTTPS is considered beneficial, as it offers an added layer of security and trust.



HTTP Requests and Responses

Each interaction between the client and server is called a message. HTTP messages are requests or responses. Client devices submit HTTP requests to servers, which reply by sending HTTP responses back to the clients.

Each HTTP request contains encoded data, with information such as:

- The specific version of HTTP followed. HTTP and HTTP/2 are the two versions.
- A URL. This points to the resource on the web.
- An HTTP method. This indicates the specific action the request expects to receive from the server in its response.
- HTTP request headers. This includes data such as what type of browser is being used and what data the request is seeking from the server. For example cookies.
- An HTTP body. This is optional information the server needs from the request, such as user forms -- username/password logins, short responses and file uploads -- that are being submitted to the website.



HTTP Responses

The HTTP response message is the data received by a client device from the web server. HTTP responses typically include the following data:

- **HTTP status code**, which indicates the status of the request to the client device. Responses may indicate success, an informational response, a redirect, or errors on the server or client side
- .
- **HTTP response headers**, which send information about the server and requested resources.
- **An HTTP body (optional)**. If a request is successful, this contains the requested data in the form of HTML code, which is translated into a web page by the client browser.



HTTP Status Codes

In response to HTTP requests, servers often issue response codes, indicating the request is being processed, there was an error in the request or that the request is being redirected. Common response codes include:

- **200 OK.** This means that the request, such as GET or POST, worked and is being acted upon.
- **300 Moved Permanently.** This response code means that the URL of the requested resource has been changed permanently.
- **401 Unauthorized.** The client, or user making the request of the server, has not been authenticated.
- **403 Forbidden.** The client's identity is known but has not been given access authorization.
- **404 Not Found.** This is the most frequent error code. It means that the URL is not recognized or the resource at the location does not exist.
- **500 Internal Server Error.** The server has encountered a situation it doesn't know how to handle.

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a web or a neural network structure.

Transmission Control Protocol (TCP)

Transmission Control Protocol (TCP) is a standard that defines how to establish and maintain a network conversation by which applications can exchange data.

TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together, TCP and IP are the basic rules that define the internet.

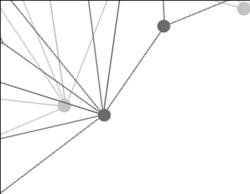
How Transmission Control Protocol works

TCP is a connection-oriented protocol, which means a connection is established and maintained until the applications at each end have finished exchanging messages.



TCP performs the following actions:

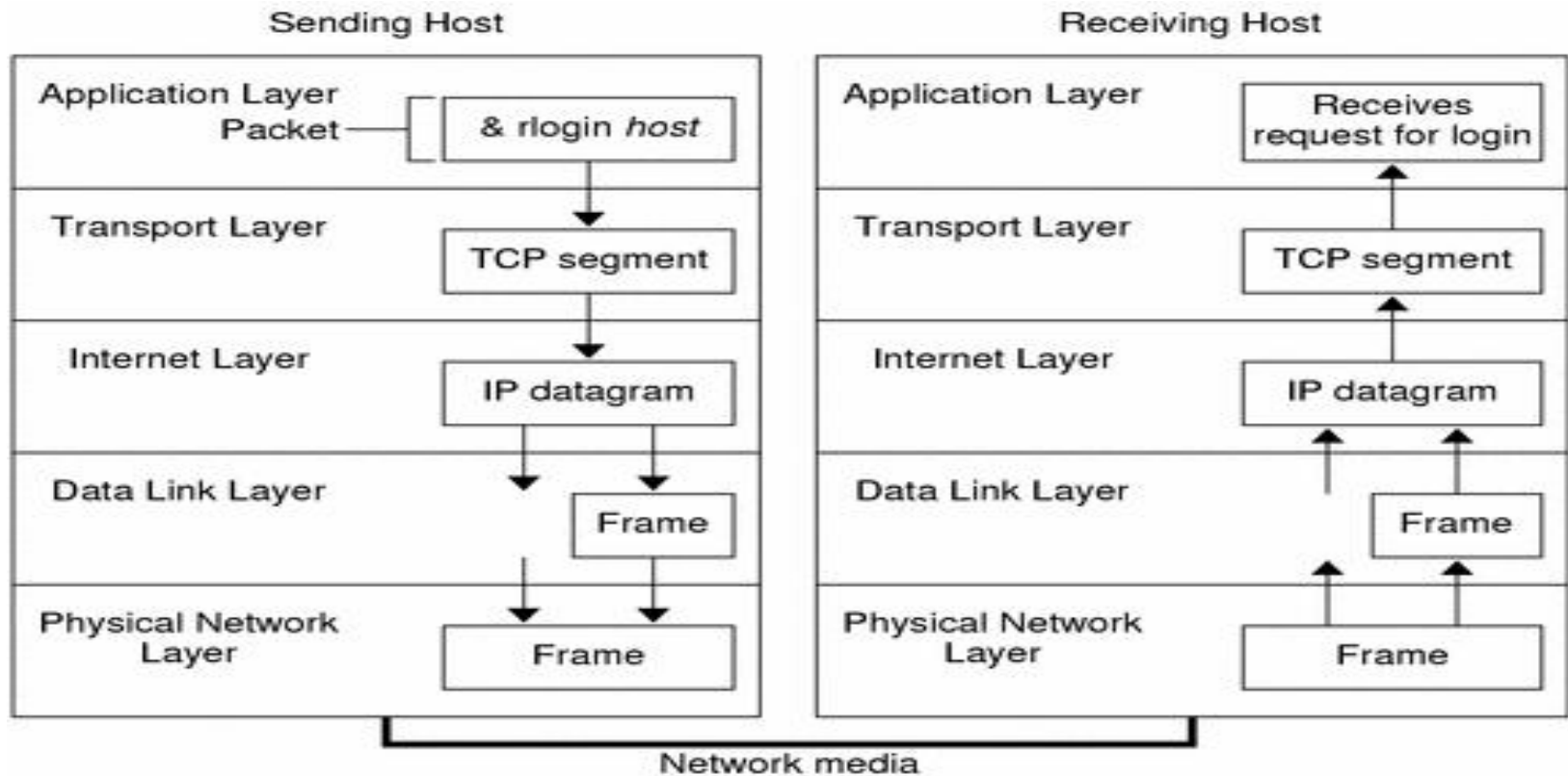
- Determines how to break application data into packets that networks can deliver;
- Sends packets to, and accepts packets from, the network layer;
- Manages flow control;
- Handles retransmission of dropped or garbled packets, as it's meant to provide error-free data transmission;
- Acknowledges all packets that arrive.



How the TCP/IP Protocols Handle Data Communications

When a user issues a command that uses a TCP/IP application layer protocol, a series of events is initiated.

- The user's command or message passes through the TCP/IP protocol stack on the local machine.
- Then the command or message passes across the network media to the protocols on the recipient.
- The protocols at each layer on the sending host add information to the original data.



How a Packet Travels Through the TCP/IP Stack



Life cycle of a packet

- The life cycle starts when you issue a command or send a message.
 - The life cycle finishes when the appropriate application on the receiving host receives the packet.
-
- **Application Layer—User Initiates Communication**
 - **Transport Layer—Data Encapsulation Begins**
 - **Internet Layer — Determines the IP addresses for the datagrams**
 - **Data-Link Layer—Framing Takes Place**
 - **Physical Network Layer—Preparing the Frame for Transmission**



Application Layer

- The packet's history begins when a user on one host sends a message or issues a command that must access a remote host.
- The application protocol formats the packet so that the appropriate transport layer protocol, TCP or UDP, can handle the packet.



Transport Layer

When the data arrives at the transport layer, the protocols at the layer start the process of data encapsulation.

TCP Segmentation

- TCP is often called a “**connection-oriented**” protocol because TCP ensures the successful delivery of data to the receiving host.
- TCP divides the data that is received from the application layer into **Segments** and attaches a header to each segment.
- **Segment headers** contain sender and recipient ports, segment ordering information, and a data field that is known as a checksum. The TCP protocols on both hosts use the checksum data to determine if the data transfers without error.



UDP Packets

- UDP is a “**connectionless**” protocol. Unlike TCP, UDP does not check that data arrived at the receiving host. Instead, UDP formats the message that is received from the application layer into UDP packets. **UDP attaches a header to each packet.**
 - The header contains the sending and receiving host ports
 - A field with the length of the packet
 - A checksum
- The sending UDP process attempts to send the packet to its peer UDP process on the receiving host. The application layer determines whether the receiving UDP process acknowledges the reception of the packet. UDP requires no notification of receipt.

A decorative graphic in the top-left corner consisting of a network of nodes and lines. It features a central node with several lines radiating from it, and other nodes connected in a web-like structure, some with small colored dots (grey, black, light blue).

Internet Layer

- TCP and UDP pass their segments and packets down to the Internet layer, where the IP protocol handles the segments and packets.
- IP prepares them for delivery by formatting them into units called IP datagrams.
- IP then determines the IP addresses for the datagrams, so that they can be delivered effectively to the receiving host.



IP Datagrams

- IP attaches an IP header to the segment or packet's header in addition to the information that is added by TCP or UDP.
- Information in the IP header includes the IP addresses of the sending and receiving hosts, datagram length, and datagram sequence order.
- This information is provided if the datagram exceeds the allowable byte size for network packets and must be fragmented.

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a partial mesh or star topology.

Data-Link Layer

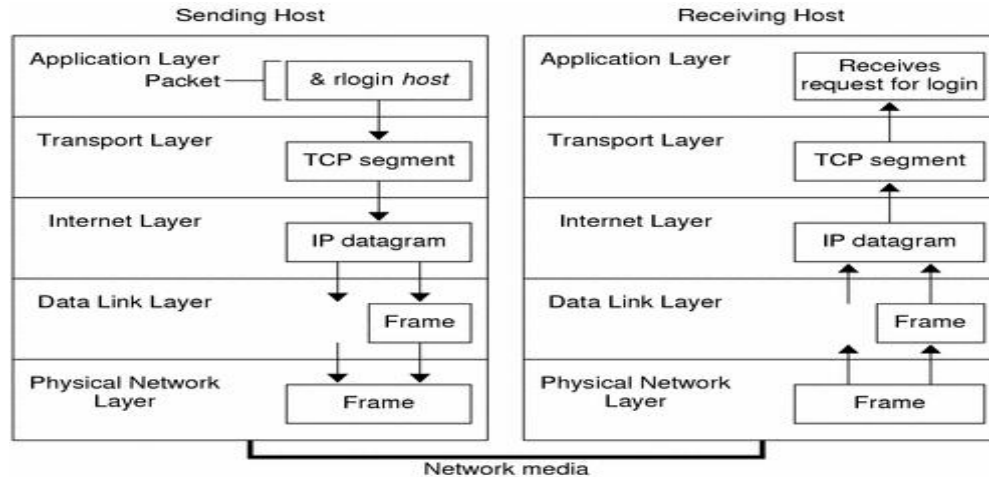
- Data-link layer protocols, such as PPP, format the IP datagram into a frame.
- These protocols attach a third header and a footer to “**frame**” the datagram.
- The frame header includes a cyclic redundancy check (CRC) field that checks for errors as the frame travels over the network media. Then the data-link layer passes the frame to the physical layer.

Physical Network Layer

The physical network layer on the sending host receives the frames and converts the IP addresses into the hardware addresses appropriate to the network media. The physical network layer then sends the frame out over the network media.

How the Receiving Host Handles the Packet

When the packet arrives on the receiving host, the packet travels through the TCP/IP protocol stack in the reverse order from that which the packet travels on the sender. Moreover, each protocol on the receiving host strips off header information that is attached to the packet by its peer on the sending host.



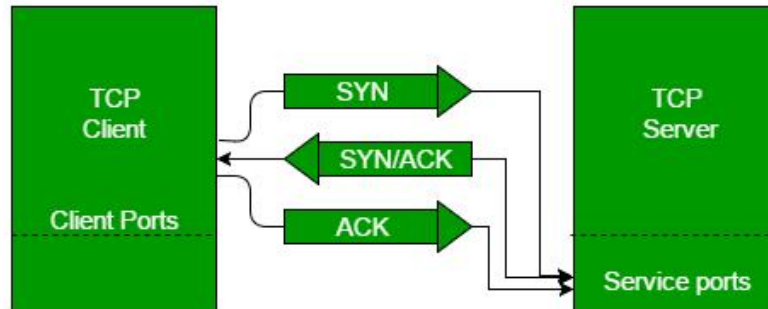


The following process occurs:

- The physical network layer receives the packet in its frame form. The physical network layer computes the CRC of the packet, then sends the frame to the data link layer.
- The data-link layer verifies that the CRC for the frame is correct and strips off the frame header and CRC. Finally, the data link protocol sends the frame to the Internet layer.
- The Internet layer reads information in the header to identify the transmission. Then Internet layer determines if the packet is a fragment. If the transmission is fragmented, IP reassembles the fragments into the original datagram. IP then strips off the IP header and passes the datagram on to transport layer protocols.
- The transport layer (TCP and UDP) reads the header to determine which application layer protocol must receive the data. Then TCP or UDP strips off its related header. TCP or UDP sends the message or stream up to the receiving application.
- The application layer receives the message. The application layer then performs the operation that the sending host requested.

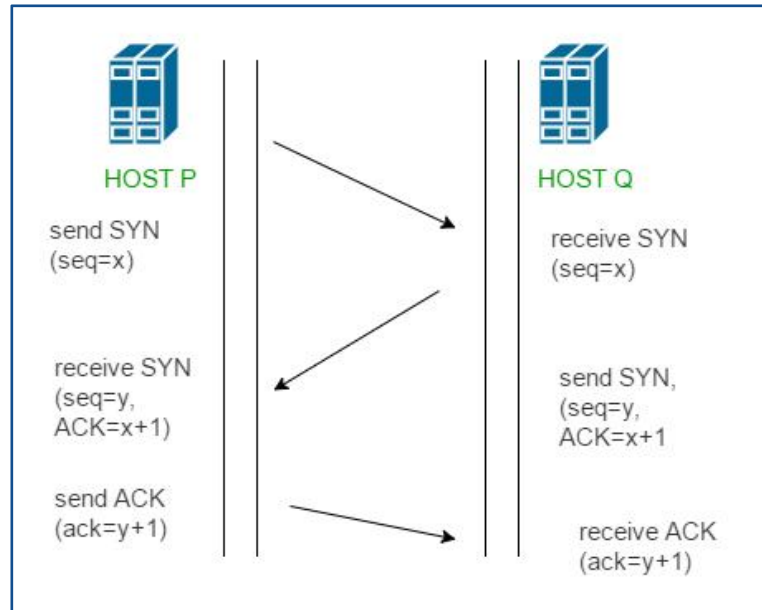
Three-Way Handshaking

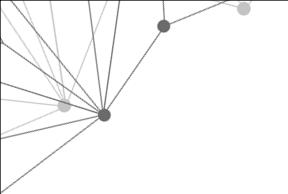
- TCP uses segments to determine whether the receiving host is ready to receive the data. When the sending TCP wants to establish connections, TCP sends a segment that is called a SYN to the TCP protocol on the receiving host.
- The receiving TCP returns a segment that is called an ACK to acknowledge the successful receipt of the segment.
- The sending TCP sends another ACK segment, then proceeds to send the data. This exchange of control information is referred to as a **three-way handshake**.



Continue...

TCP provides reliable communication with something called **Positive Acknowledgement with Re-transmission(PAR)**. There are three segments are exchanged between sender(client) and receiver(server) for a reliable TCP connection to get established. let us see the detailed process.



- 
- A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a web or a neural network structure.
- **Step 1 (SYN):** In the first step, the client wants to establish a connection with a server, so it sends a segment with SYN(Synchronize Sequence Number) which informs the server that the client is likely to start communication and with what sequence number it starts segments with.
 - **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of the segment it received and SYN signifies with what sequence number it is likely to start the segments with.
 - **Step 3 (ACK):** In the final part client acknowledges the response of the server and they both establish a reliable connection with which they will start the actual data transfer



Flow Control

Flow control is the management of data flow between computers or devices or between nodes in a network so that the data can be handled at an efficient pace.

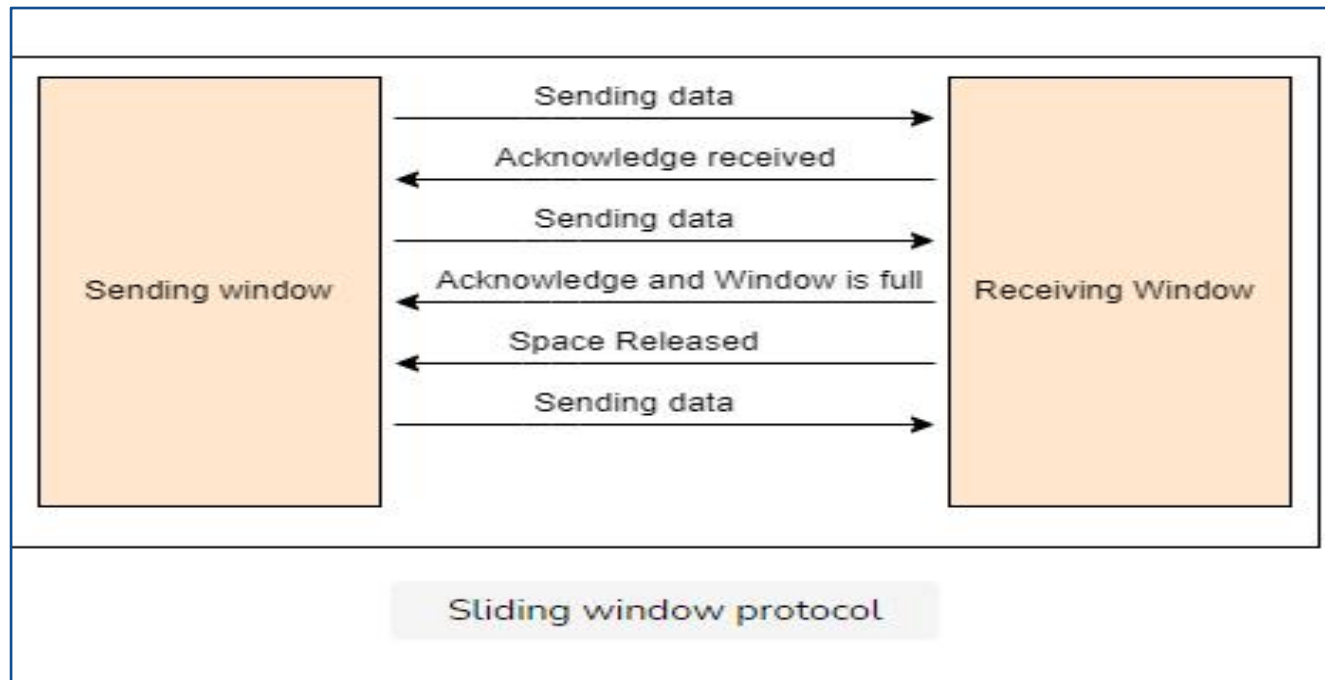
Too much data arriving before a device can handle it causes data overflow, meaning the data is either lost or must be retransmitted.

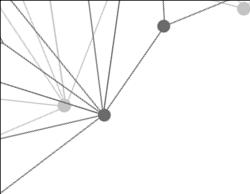
In order to achieve this, the TCP protocol uses a mechanism called the **Sliding Window Protocol**.

A decorative graphic in the top-left corner consisting of a network of nodes and connecting lines, resembling a partial graph or a stylized molecular structure.

The sliding window protocol

- In the sliding window protocol method, when we are establishing a connection between sender and receiver, there are two buffers created. Each of these two buffers are assigned to the sender, called the sending window, and to the receiver, called the receiving window.
- When the sender sends data to the receiver, the receiving window sends back the remaining receiving buffer space. As a result, the sender cannot send more data than the available receiving buffer space.



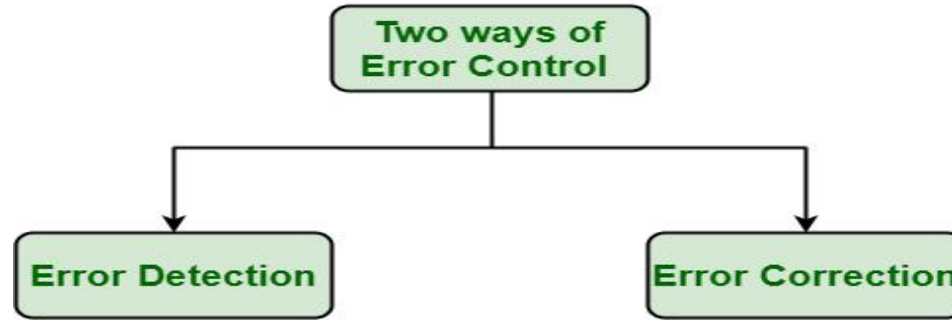


- In this example, the sending window sends data to the receiving window. The receiving window sends the acknowledgment after receiving the data and then the sending window sends another data frame.
- However, this time, along with the received acknowledgment, the receiving window also sends another message saying that the available memory is full.
- The sending window pauses the transmission of data until it gets the acknowledgment of the receiving window that space has been released and it can continue the transmission process.



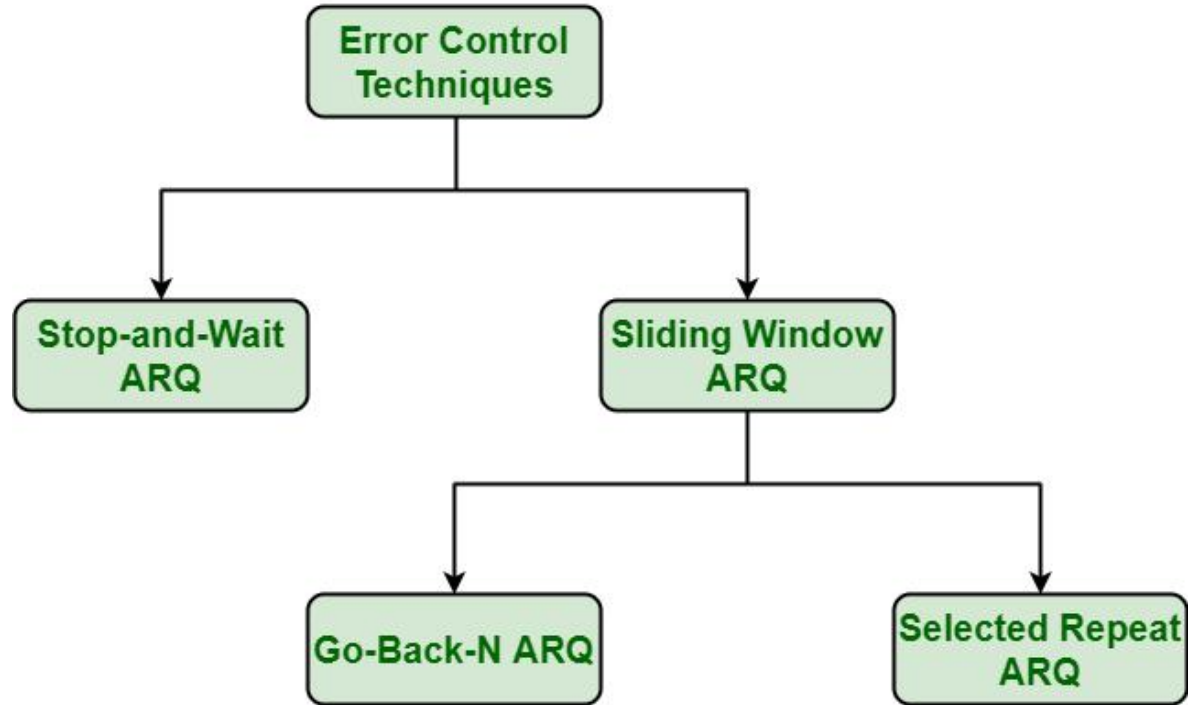
Error Control in Data Link Layer

- Error control is a process in data link layer for detecting or identifying and re-transmitting data frames that might be lost or corrupted during transmission.
- The Data-link layer follows technique known as re-transmission of frames to detect or identify transit errors and also to take necessary actions that are required to reduce or remove such errors.
- Each and every time an error is detected during transmission, particular data frames are retransmitted and this process is known as ARQ (Automatic Repeat Request).



Error Detection : Error detection means detection or identification of errors. These errors may cause due to noise or any other impairments during transmission from transmitter to the receiver, in communication system. It is class of technique for detecting garbled i.e. unclear and distorted data or message.

Error Correction : Error correction, as name suggests, simply means correction or solving or fixing of errors. It simply means reconstruction and rehabilitation of original data that is error-free. But error correction method is very costly and is very hard.



Various Techniques for Error Control

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a partial graph or a stylized molecular structure.

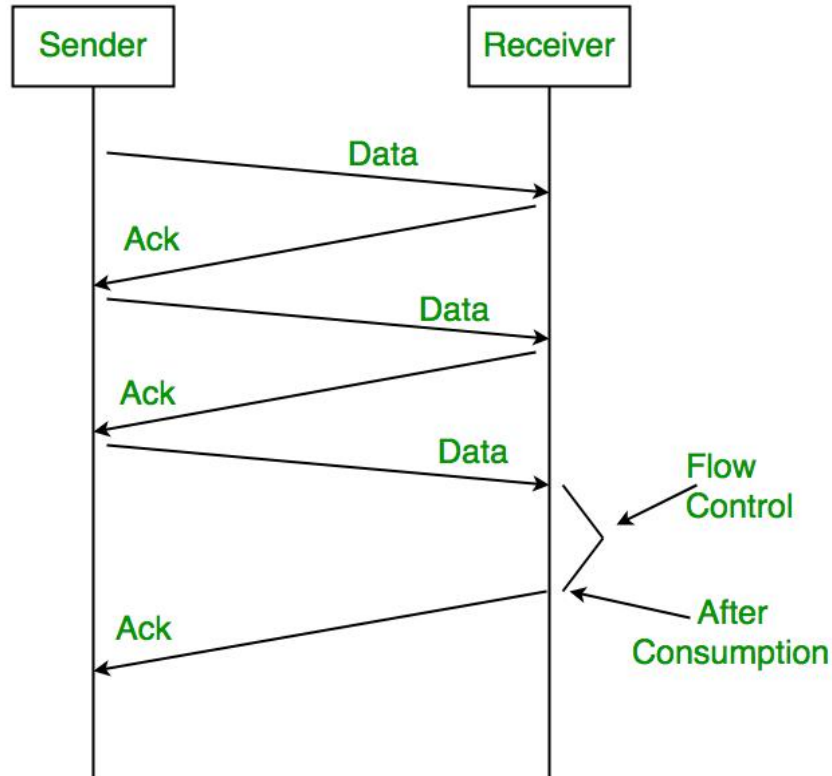
Stop-and-Wait/ARQ

Stop-and-Wait ARQ is also known as alternating bit protocol. It is one of simplest flow and error control techniques or mechanisms. This mechanism is generally required in telecommunications to transmit data or information among two connected devices.

Characteristics

- Used in Connection-oriented communication.
- It offers error and flows control
- It is used in Data Link and Transport Layers
- Stop and Wait for ARQ mainly implements the Sliding Window Protocol concept with Window Size 1

Simple Stop and Wait





Sender:

Rule 1) Send one data packet at a time.

Rule 2) Send the next packet only after receiving acknowledgement for the previous.

Receiver:

Rule 1) Send acknowledgement after receiving and consuming a data packet.

Rule 2) After consuming packet acknowledgement need to be sent (Flow Control)

Problems :

1. Lost Data
2. Lost Acknowledgement:
3. Delayed Acknowledgement/Data:



Stop and Wait for ARQ (Automatic Repeat Request)

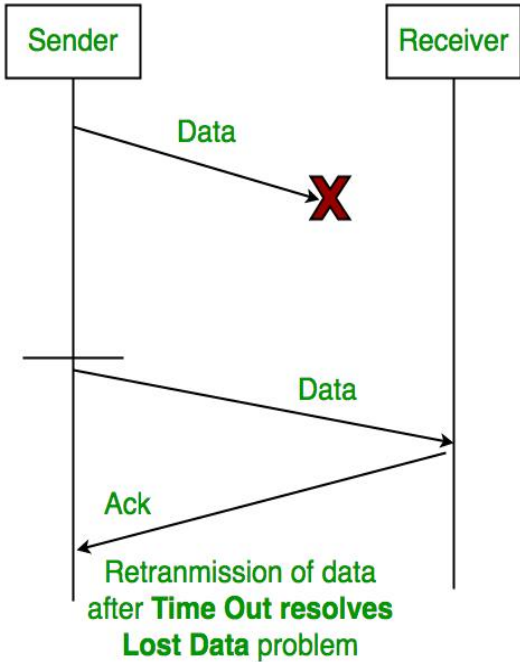
These problems are resolved by Stop and Wait for ARQ (Automatic Repeat Request) that does both error control and flow control.

Stop (and) Wait + Time Out + Sequence No.(Data) + Sequence No. (ACK)

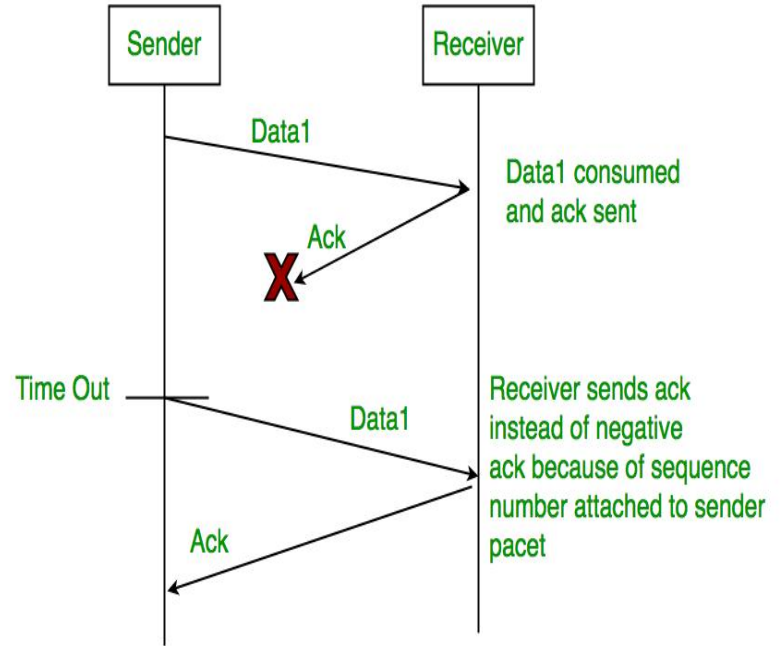
↑
Lost Data

↑
Lost Ack

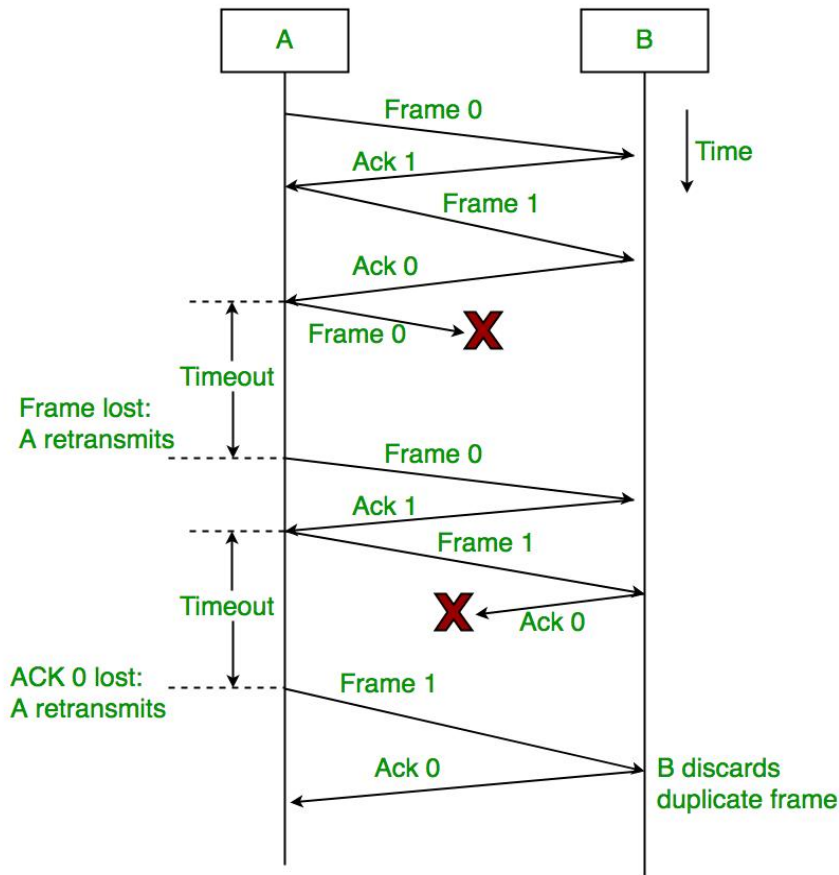
↑
Delayed Ack



Time Out



Sequence Number (Data)



Delayed Acknowledgement can be resolved by introducing sequence numbers for acknowledgement also.



Working of Stop and Wait for ARQ:

- 1) Sender A sends a data frame or packet with sequence number 0.
- 2) Receiver B, after receiving the data frame, sends an acknowledgement with sequence number 1 (the sequence number of the next expected data frame or packet)
- 3) There is only a one-bit sequence number that implies that both sender and receiver have a buffer for one frame or packet only.

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a partial graph or a network topology.

Characteristics of Stop and Wait ARQ:

- It uses a link between sender and receiver as a half-duplex link
- Throughput = 1 Data packet/frame per RTT (Round Trip Time)
- If the **Bandwidth*Delay product** is very high, then the stop and wait for protocol is not so useful. The sender has to keep waiting for acknowledgements before sending the processed next packet.
- It is an example of “**Closed Loop OR connection-oriented**” protocols
- It is a special category of **SWP** where its window size is 1
- Irrespective of the number of packets sender is having stop and wait for protocol requires only 2 sequence numbers 0 and 1



Sliding Window ARQ

This technique is generally used for continuous transmission error control.

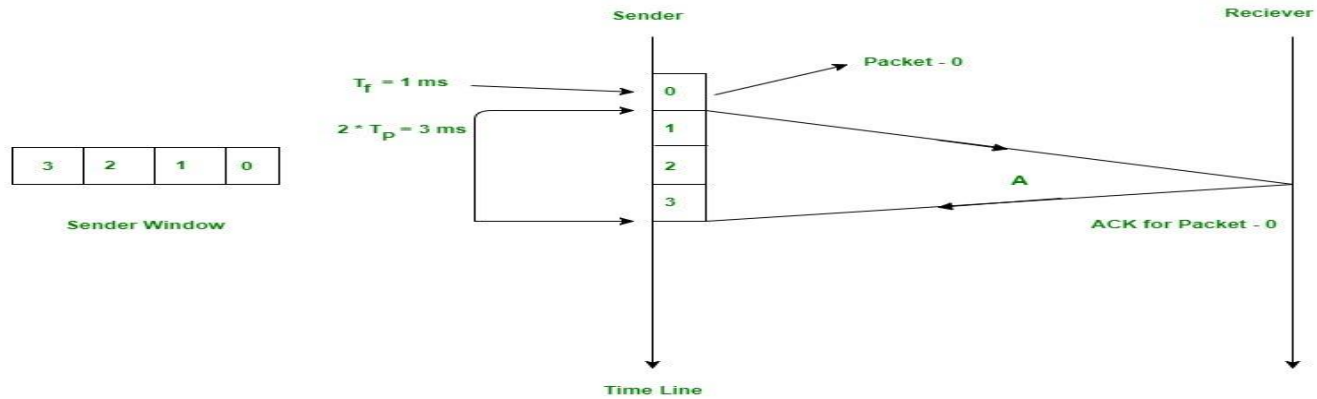
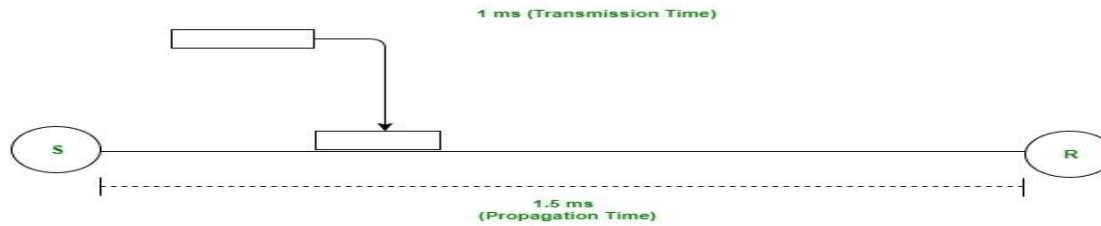
It is further categorized into two categories as given below :

- **Go-Back-N ARQ**
- **Selective Repeat ARQ**

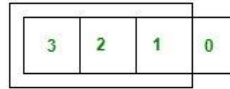


Go-Back-N ARQ

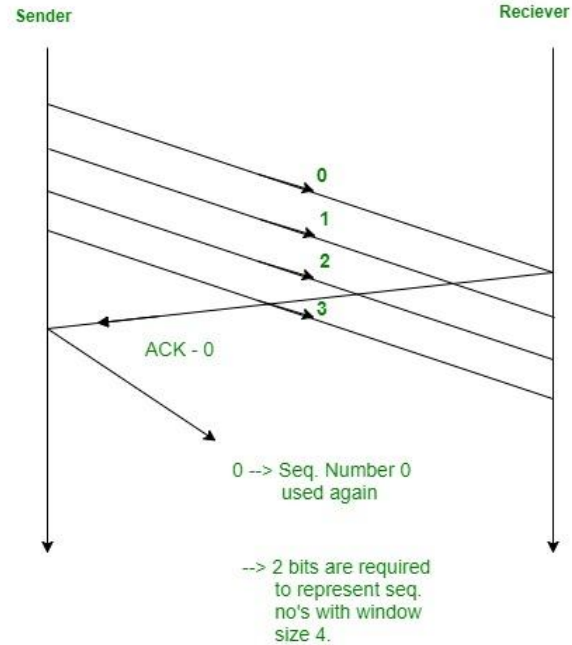
Go-Back-N ARQ is form of ARQ protocol in which transmission process continues to send or transmit total number of frames that are specified by window size even without receiving an ACK (Acknowledgement) packet from the receiver. It uses sliding window flow control protocol. If no errors occur, then operation is identical to sliding window.



In the picture given above, after sender has transmitted packet 0, it will immediately transmit packets 1, 2, 3. Acknowledgement for 0 will arrive after $2 * 1.5 = 3 \text{ ms}$. In Stop and Wait, in time $1 + 2 * 1.5 = 4 \text{ ms}$, we were transferring one packet only. Here we keep a window of packets that we have transmitted **but not yet acknowledged**.

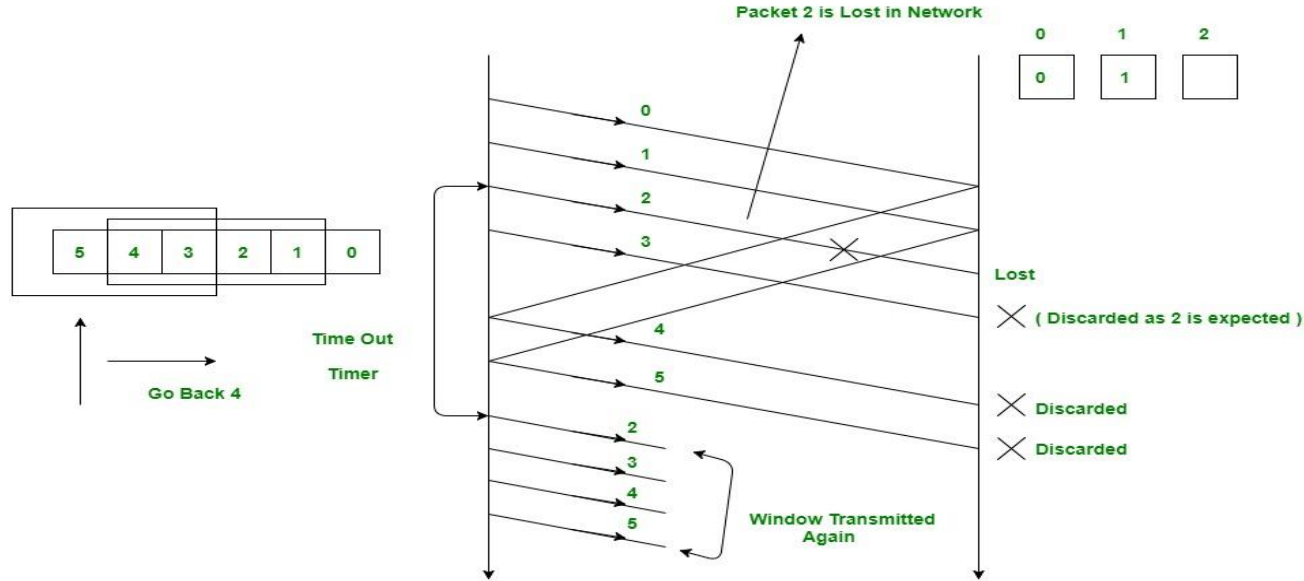


Window Slided On
Receiving Ack of
Packet - 0



After we have received the Ack for packet 0, window slides and the next packet can be assigned sequence number 0. We reuse the sequence numbers which we have acknowledged so that header size can be kept minimum as shown in the diagram given above

Go Back N (GBN) Protocol Cont..



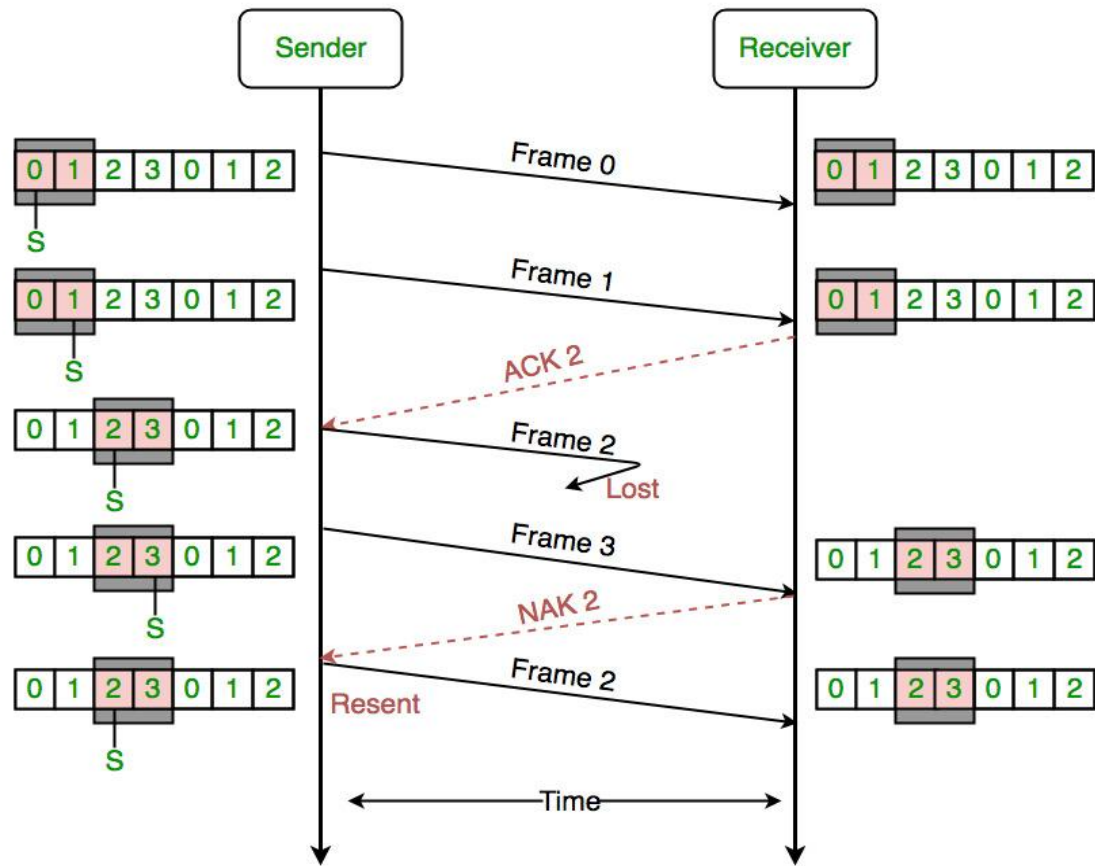
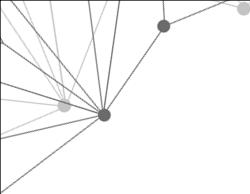
Suppose the sender has sent the packets 0, 1, 2 and 3. After acknowledging the packets 0 and 1, receiver is now expecting packet 2 and sender window has also slid to further transmit the packets 4 and 5. Now suppose the packet 2 is lost in the network, Receiver will discard all the packets which sender has transmitted after packet 2 as it is expecting sequence number of 2. On the sender side for every packet send there is a time out timer which will expire for packet number 2. Now from the last transmitted packet 5 sender will go back to the packet number 2 in the current window and transmit all the packets till packet number 5. **That's why it is called Go Back N.**

A decorative graphic in the top-left corner consisting of a network of nodes and connecting lines, resembling a partial mesh or star topology.

Selective Repeat Protocol (SRP)

This protocol(SRP) is mostly identical to GBN protocol, except that buffers are used and the receiver, and the sender, each maintains a window of same size.

- SRP works better when the link is very unreliable.
- SRP also requires full-duplex link.
- Sender's Windows (W_s) = Receiver's Windows (W_r).
- Window size should be less than or equal to half the sequence number in SR protocol.
- Sender can transmit new packets as long as their number is with W of all unACKed packets.
- Sender retransmit un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.
- Receiver ACKs all correct packets.





Congestion Control

Congestion Control is a mechanism that controls the entry of data packets into the network, enabling a better use of a shared network infrastructure and avoiding congestive collapse.

- **Congestive-Avoidance Algorithms (CAA)** are implemented at the TCP layer as the mechanism to avoid congestive collapse in a network.
- Congestion-avoidance algorithms work by detecting congestion and adjusting the data transmission rate to avoid it. This ensures the network is usable by everyone.
- Congestion control occurs at different parts of the network, from Active Queue Management that reorders packets in the Network Interface Controller (NIC) to variations of Random Early Detection in routers.



Some terminologies we must aware of..

- **Maximum Segment Size (MSS):** A property of the TCP layer, MSS is the maximum size of a payload that can be sent in a single data packet. This size does not include the header size.
- **Maximum Transmission Unit (MTU):** The maximum size of a payload including the headers that can be sent in a single packet. This is a property of the Data Link layer. The difference from MSS is that if a packet exceeds MTU, it is broken into multiple chunks obeying the MSS of the link. However, if a packet exceeds the MSS, it is dropped altogether.
- **Cwnd (Congestion Window):** The number of unacknowledged packets (MSS) at any given moment that can be in transit. The congestion window increases, decreases, or stays the same depending on how many of the initial packets were acknowledged and how long it took to do so.
- **Initcwnd (Initial Congestion Window):** The initial value of cwnd. Usually, algorithms start with a small multiple of MSS and increase sharply.



TCP Connection Lifecycle

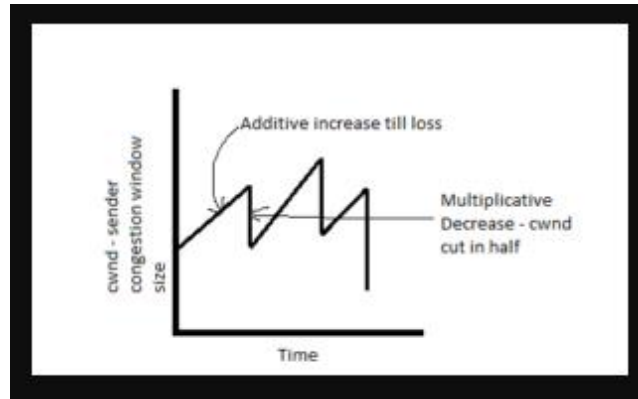
When a connection is established, the sender does not immediately overwhelm the network; instead, it starts slow and then adjusts according to the network bandwidth.

In TCP, the congestion-avoidance mechanisms kick in when the network detects loss because TCP perceives every loss as an event due to congestion. There are two ways in which TCP assumes that packets are getting lost:

- When there is a timeout
- When the server receives three duplicate ACKs for a data packet

Additive Increase Multiplicative Decrease (AIMD)

In this approach, senders increase the cwnd by 1 MSS on every successful ACK (additive increase). On the detection of loss, the cwnd is cut in half (multiplicative decrease). This shows a saw-tooth behavior in cwnd.





Slow Start

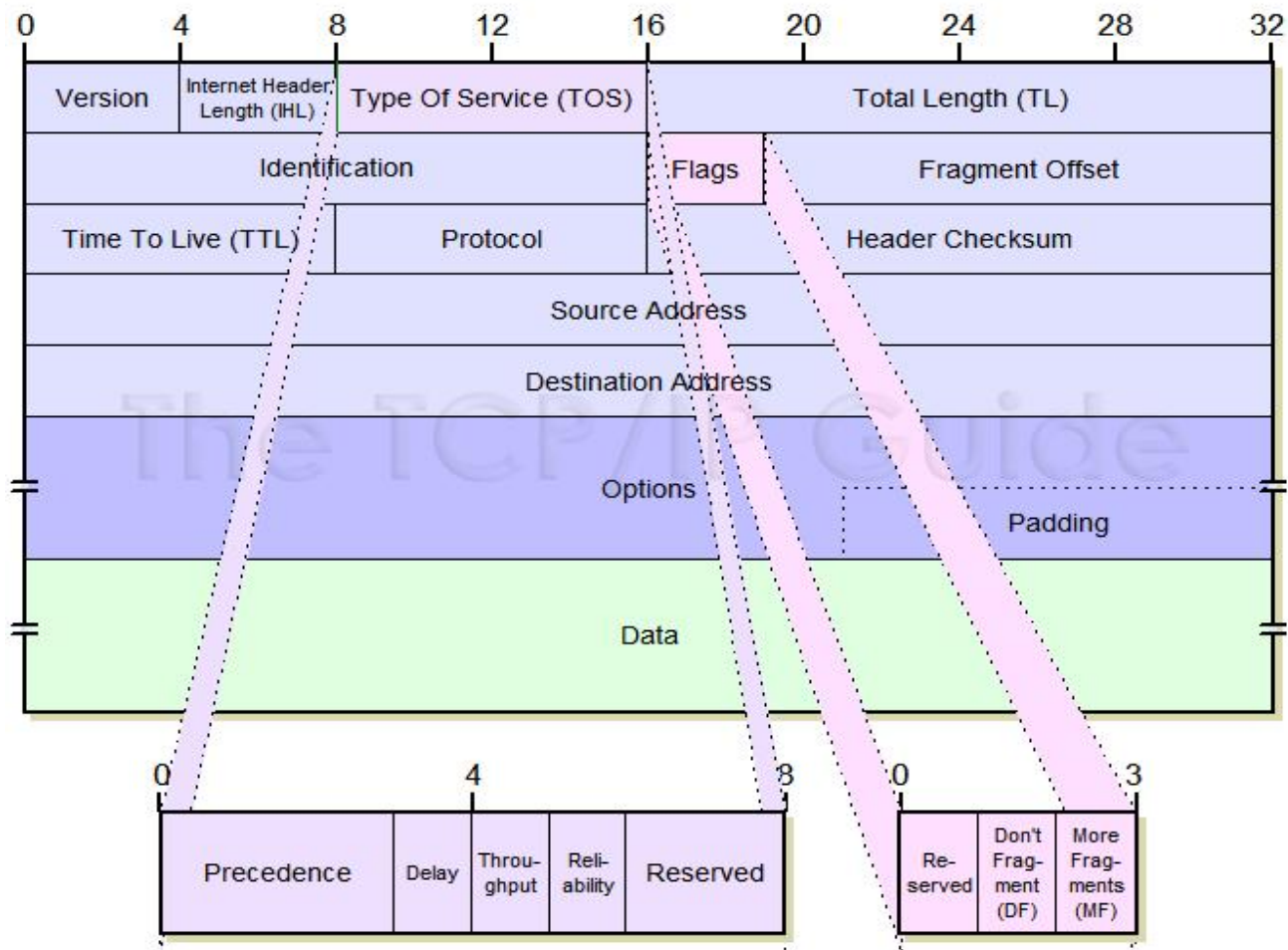
Contrary to its name, this algorithm starts with `initcwnd` set to 1 and doubles the `cwnd` after every successful ACK until it reaches the `ssthresh` (Slow Start threshold), after which it increases the `cwnd` linearly by 1 MSS on every ACK.

When a loss is detected, the `ssthresh` is set to one-half of the `cwnd` at that time, and `cwnd` is decreased.


A decorative graphic in the top-left corner consisting of a network of nodes and connecting lines, resembling a partial mesh or star topology.

IP Datagram General Format


- Data transmitted over an internet using IP is carried in messages called IP datagrams. Like all network protocol messages, IP uses a specific format for its datagrams.
- The IPv4 datagram is conceptually divided into two pieces: the **header** and the **payload**. The header contains addressing and control fields, while the payload carries the actual data to be sent over the internetwork. Unlike some message formats, IP datagrams do not have a footer following the payload.
- Even though IP is a relatively simple, connectionless, “unreliable” protocol, the IPv4 header carries a fair bit of information, which makes it rather large. At a minimum, it is 20 bytes long, and with options can be significantly longer.



Internet Protocol Version 4 (IPv4) Datagram Format



Field Name	Size (bytes)	Description
Version	1/2 (4 bits)	Version: Identifies the version of IP used to generate the datagram. For IPv4, this is of course the number 4. The purpose of this field is to ensure compatibility between devices that may be running different versions of IP.
IHL	1/2 (4 bits)	Internet Header Length (IHL): Specifies the length of the IP header, in 32-bit words. This includes the length of any options fields and padding.
TOS	1	Type Of Service (TOS): A field designed to carry information to provide quality of service features, such as prioritized delivery, for IP datagrams. (DS).
TL	2	Total Length (TL): Specifies the total length of the IP datagram, in bytes. Since this field is 16 bits wide, the maximum length of an IP datagram is 65,535 bytes, though most are much smaller.
Identification	2	This field contains a 16-bit value that is common to each of the fragments belonging to a particular message; for datagrams originally sent unfragmented it is still filled in, so it can be used if the datagram must be fragmented by a router during delivery. This field is used by the recipient to reassemble messages without accidentally mixing fragments from different messages.



Field Name	Size (bytes)	Description
Flags	3/8 (3 bits)	Three Control Flags, two of which are used to control fragmentation, another is reserved. <ul style="list-style-type: none">• Reserved (Not used)• DF (Dont Fragment)• MF (More fragment)
Fragment Offset	1 5/8 (13 bits)	Fragment Offset: When fragmentation of a message occurs, this field specifies the offset, or position, in the overall message where the data in this fragment goes. It is specified in units of 8 bytes (64 bits). The first fragment has an offset of 0.
TTL	1	Time To Live (TTL): Specifies how long the datagram is allowed to “live” on the network, in terms of router hops. Each router decrements the value of the TTL field (reduces it by one) prior to transmitting it. If the TTL field drops to zero, the datagram is assumed to have taken too long a route and is discarded.
Header Checksum	2	Header Checksum: A checksum computed over the header to provide basic protection against corruption in transmission. It's just a 16-bit checksum. It is calculated by dividing the header bytes into words (a word is two bytes) and then adding them together. The data is not checksummed, only the header.



Field Name	Size (bytes)	Description																																	
Protocol	1	<p>Protocol: Identifies the higher-layer protocol (generally either a transport layer protocol or encapsulated network layer protocol) carried in the datagram.</p> <table><tr><th>Value (Hexadecimal)</th><th>Value (Decimal)</th><th>Protocol</th></tr><tr><td>00</td><td>0</td><td>Reserved</td></tr><tr><td>01</td><td>1</td><td>ICMP</td></tr><tr><td>02</td><td>2</td><td>IGMP</td></tr><tr><td>03</td><td>3</td><td>GGP</td></tr><tr><td>04</td><td>4</td><td>IP-in-IP Encapsulation</td></tr><tr><td>06</td><td>6</td><td>TCP</td></tr><tr><td>08</td><td>8</td><td>EGP</td></tr><tr><td>11</td><td>17</td><td>UDP</td></tr><tr><td>32</td><td>50</td><td>Encapsulating Security[®] Payload (ESP) Extension Header</td></tr><tr><td>33</td><td>51</td><td>Authentication Header (AH) Extension Header</td></tr></table>	Value (Hexadecimal)	Value (Decimal)	Protocol	00	0	Reserved	01	1	ICMP	02	2	IGMP	03	3	GGP	04	4	IP-in-IP Encapsulation	06	6	TCP	08	8	EGP	11	17	UDP	32	50	Encapsulating Security [®] Payload (ESP) Extension Header	33	51	Authentication Header (AH) Extension Header
Value (Hexadecimal)	Value (Decimal)	Protocol																																	
00	0	Reserved																																	
01	1	ICMP																																	
02	2	IGMP																																	
03	3	GGP																																	
04	4	IP-in-IP Encapsulation																																	
06	6	TCP																																	
08	8	EGP																																	
11	17	UDP																																	
32	50	Encapsulating Security [®] Payload (ESP) Extension Header																																	
33	51	Authentication Header (AH) Extension Header																																	
Source Address	4	<p>The 32-bit IP address of the originator of the datagram. Intermediate devices such as routers may handle the datagram; they do not normally put their address into this field—it is always the device that originally sent the datagram.</p>																																	

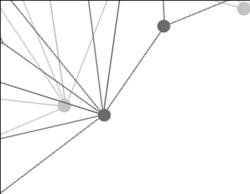


Field Name	Size (bytes)	Description
Destination Address	4	Destination Address: The 32-bit IP address of the intended recipient of the datagram. Again, even though devices such as routers may be the intermediate targets of the datagram, this field is always for the ultimate destination.
Options	Variable	Options: One or more of several types of options may be included after the standard headers in certain IP datagrams.
Padding	Variable	Padding: If one or more options are included, and the number of bits used for them is not a multiple of 32, enough zero bits are added to “pad out” the header to a multiple of 32 bits (4 bytes).
Data	Variable	The data to be transmitted in the datagram, either an entire higher order message or a fragmented one.



IPV4 VS IPV6

IPV4	IPV6
IPv4 has a 32-bit address length	IPv6 has a 128-bit address length
It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
In IPv4 end to end, connection integrity is Unachievable	In IPv6 end to end, connection integrity is Achievable
It can generate 4.29×10^9 address space	Address space of IPv6 is quite large it can produce 3.4×10^{38} address space
The Security feature is dependent on application	IPSEC is an inbuilt security feature in the IPv6 protocol
Address representation of IPv4 is in decimal	Address Representation of IPv6 is in hexadecimal



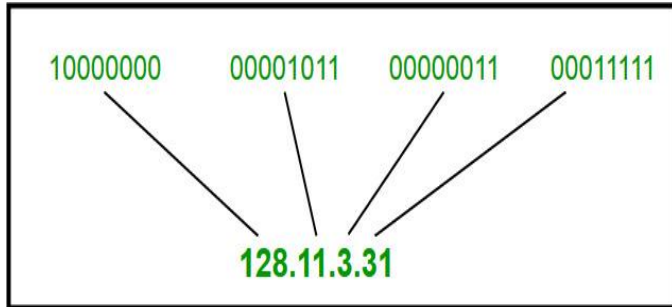
IPV4	IPV6
Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation performed only by the sender
In IPv4 Packet flow identification is not available	In IPv6 packet flow identification are Available and uses the flow label field in the header
In IPv4 checksum field is available	In IPv6 checksum field is not available
It has broadcast Message Transmission Scheme	In IPv6 multicast and anycast message transmission scheme is available
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided
IPv4 has a header of 20-60 bytes.	IPv6 has header of 40 bytes fixed
IPv4 consist of 4 fields which are separated by dot (.)	IPv6 consist of 8 fields, which are separated by colon (:)

Classful IP Addressing

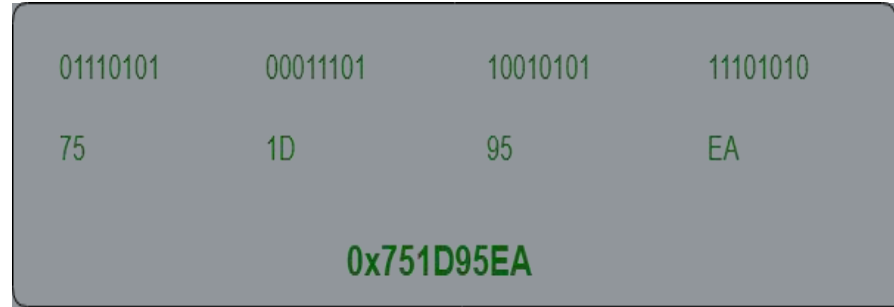
IP address is an address having information about how to reach a specific host, especially outside the LAN. An IP address is a 32 bit unique address having an address space of 2^{32} .

Generally, there are two notations in which IP address is written, dotted decimal notation and hexadecimal notation.

- Dotted Decimal Notation
- Hexadecimal Notation



Dotted Decimal Notation



Hexadecimal Notation

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a web or a neural network structure.

Classful Addressing

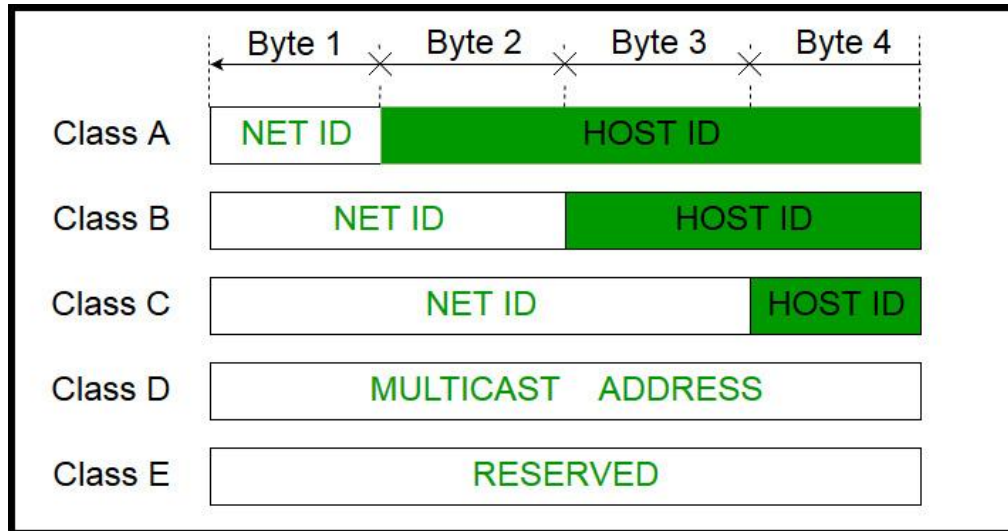
The 32 bit IP address is divided into five sub-classes. These are:

- **Class A**
 - **Class B**
 - **Class C**
 - **Class D**
 - **Class E**
-
- Each of these classes has a valid range of IP addresses.
 - Classes D and E are reserved for multicast and experimental purposes respectively.
 - The order of bits in the first octet determine the classes of IP address.

IPv4 address is divided into two parts:

- Network ID
- Host ID

The class of IP address is used to determine the bits used for network ID and host ID and the number of total networks and hosts possible in that particular class.





Class A:

IP address belonging to class A are assigned to the networks that contain a large number of hosts.

- The network ID is 8 bits long.
- The host ID is 24 bits long.



Class A

Class B:

IP address belonging to class B are assigned to the networks that ranges from medium-sized to large-sized networks.

- The network ID is 16 bits long.
- The host ID is 16 bits long.



Class B



Class C:

IP address belonging to class C are assigned to small-sized networks.

- The network ID is 24 bits long.
- The host ID is 8 bits long.



Class C

Class D:

IP address belonging to class D are reserved for multi-casting. The higher order bits of the first octet of IP addresses belonging to class D are always set to 1110. The remaining bits are for the address that interested hosts recognize.

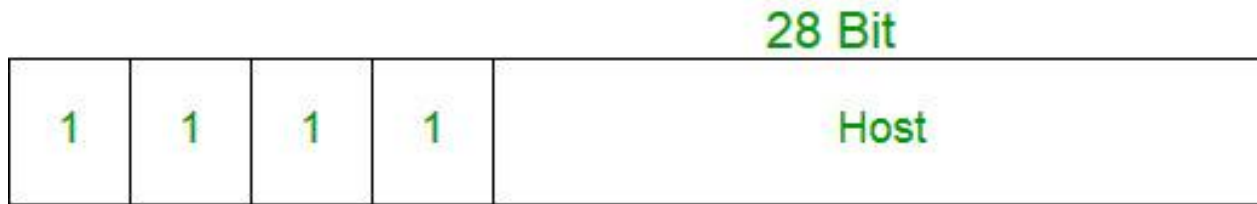
Class D does not possess any sub-net mask. IP addresses belonging to class D range from 224.0.0.0 – 239.255.255.255.



Class D

Class E:

- IP addresses belonging to class E are reserved for experimental and research purposes.
- IP addresses of class E ranges from 240.0.0.0 – 255.255.255.254.
- This class doesn't have any sub-net mask. The higher order bits of first octet of class E are always set to 1111.



Class E



Range of special IP addresses:

169.254.0.0 – 169.254.0.16 : Link local addresses

127.0.0.0 – 127.0.0.8 : Loop-back addresses

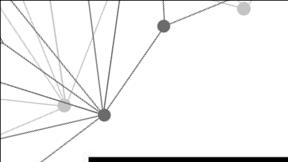
0.0.0.0 – 0.0.0.8 : used to communicate within the current network.

Rules for assigning Host ID:

- Within any network, the host ID must be unique to that network.
- Host ID in which all bits are set to 0 cannot be assigned because this host ID is used to represent the network ID of the IP address.
- Host ID in which all bits are set to 1 cannot be assigned because this host ID is reserved as a broadcast address to send packets to all the hosts present on that particular network.

Rules for assigning Network ID:

- The network ID cannot start with 127 because 127 belongs to class A address and is reserved for internal loop-back functions.
- All bits of network ID set to 1 are reserved for use as an IP broadcast address and therefore, cannot be used.
- All bits of network ID set to 0 are used to denote a specific host on the local network and are not routed and therefore, aren't used.



CLASS	LEADING BITS	NET ID BITS	HOST ID BITS	NO. OF NETWORKS	ADDRESSES PER NETWORK	START ADDRESS	END ADDRESS
CLASS A	0	8	24	2^7 (128)	2^{24} (16,777,216)	0.0.0.0	127.255.255.255
CLASS B	10	16	16	2^{14} (16,384)	2^{16} (65,536)	128.0.0.0	191.255.255.255
CLASS C	110	24	8	2^{21} (2,097,152)	2^8 (256)	192.0.0.0	223.255.255.255
CLASS D	1110	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	224.0.0.0	239.255.255.255
CLASS E	1111	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	240.0.0.0	255.255.255.255

Summary of Classful addressing



Classless Addressing

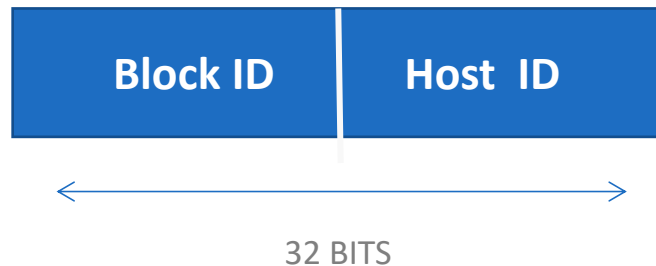
Classless addressing is an IPv4 addressing architecture that uses variable-length subnet masking.

The solution would come in 1993, as **Classless Inter-Domain Routing (CIDR)** introducing the concept of classless addressing. You see, with classful addressing, the size of networks is fixed. Each address range has a default subnet mask. Classless addressing, however, decouples IP address ranges from a default subnet mask, allowing for variable-length subnet masking (VLSM).

Using classless addressing and VLSM, addresses can be allocated much more efficiently. This is because network admins get to pick network masks, and in turn, blocks of IP addresses that are the right size for any purpose.

Lets understand it clearly

- No concept of class
- Only cocept of blocks.
- There is no default subnetting mask.



NOTATION

X.Y.Z.W/N

200.10.20.40/28
(for example)

N is the mask/ number of bits represent the block/network



Our example

200.10.20.40/28

- The the no. hosts will be $32-28 = 4$ ie. $2^4=16$
- Now the subnet mask will be 11111111.11111111.11111111.11110000 (255.255.255.240)
- We can easily find our network id by AND ing with the subnet mask
- $200.10.20.40 \text{ AND } 255.255.255.240 = 200.10.20.32$ (Network ID)(First id in the Network)
- 200.10.20.48 is the last IP in the network.



Rules for class less addressing

- IP addresses has to be contiguous
- No. of addresses in a block must be in power of 2
- The first IP address has to be perfectly divisible by block size.

Advantages

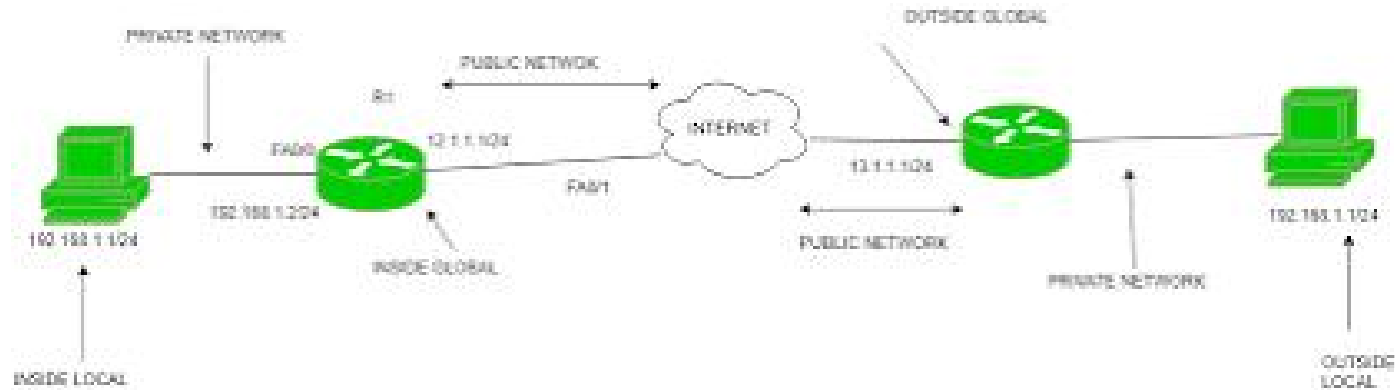
More IP address allocations. It will resolve the problem of IP wastage as classless addressing was used as a medium-term solution to help us stretch the life of IPv4.

More balanced use of IP address ranges. Classless addressing decoupled the relationship between network size and IP address and allowed for balanced use across what used to be the Class A, B, and C ranges. Far less wasted addresses.

More efficient routing. VLSM and subnetting make route aggregation and classless routing protocols possible. With route aggregation (sometimes called route summarization or supernetting), routing tables can be smaller, reducing resource consumption on routers, and saving bandwidth.

What Is NAT?

NAT stands for **network address translation**. It's a way to map multiple local private addresses to a public one before transferring the information. Organizations that want multiple devices to employ a single IP address use NAT, as do most home routers.





How Does NAT Work?

Let's say that there is a laptop connected to a home router. Someone uses the laptop to search for directions to their favorite restaurant. The laptop sends this request in a packet to the router, which passes it along to the web. But first, the router changes the outgoing IP address from a private local address to a public address.

If the packet keeps a private address, the receiving server won't know where to send the information back to — this is akin to sending physical mail and requesting return service but providing a return address of anonymous. By using NAT, the information will make it back to the laptop using the router's public address, not the laptop's private one.



NAT Types

There are three different types of NATs. People use them for different reasons, but they all still work as a NAT.

1. Static NAT

When the local address is converted to a public one, this NAT chooses the same one. This means there will be a consistent public IP address associated with that router or NAT device.

2. Dynamic NAT

Instead of choosing the same IP address every time, this NAT goes through a pool of public IP addresses. This results in the router or NAT device getting a different address each time the router translates the local address to a public address.

3. PAT

PAT stands for port address translation. It's a type of dynamic NAT, but it bands several local IP addresses to a singular public one. Organizations that want all their employees' activity to use a singular IP address use a PAT, often under the supervision of a network administrator.



Advantages:

- The internet has exploded, and while not all 7 billion people on the planet access the internet regularly, those that do often have multiple connected devices: phones, personal desktop, work laptop, tablet, TV, even refrigerators.
- Therefore, the number of devices accessing the internet far surpasses the number of IP addresses available. Routing all of these devices via one connection using NAT helps to consolidate multiple private IP addresses into one public IP address. This helps to keep more public IP addresses available even while private IP addresses proliferate.

A decorative graphic in the top-left corner consisting of a network of nodes and connecting lines, resembling a web or a neural network structure.

IP Masquerading

IP masquerading is a process where one computer acts as an IP gateway for a network. All computers on the network send their IP packets through the gateway, which replaces the source IP address with its own address and then forwards it to the internet. The source IP port number is also replaced with another port number. All hosts on the internet see the packet as originating from the gateway.

Any host on the Internet which wishes to send a packet back, ie in reply, must necessarily address that packet to the gateway. The gateway is the only host seen on the internet. The gateway rewrites the destination address, replacing its own address with the IP address of the machine which is being masqueraded, and forwards that packet on to the local network for delivery.

A decorative graphic in the top-left corner consisting of a network of nodes and connecting lines, resembling a partial mesh or star topology.

IP Masq is available on Linux and a few ISDN routers such or as the Zytel Prestige128, Cisco 770, NetGear ISDN routers, etc.

- Pro:**
- + Only (1) IP address needed (cheap)
 - + Doesn't require special application support
 - + Uses firewall software so your network can become more secure
- Con:**
- Requires a Linux box or special ISDN router (though other products might have this..)
 - Incoming traffic cannot access your internal LAN unless the internal LAN initiates the traffic or specific port forwarding software is installed. Many NAT servers CANNOT provide this functionality.
 - Special protocols need to be uniquely handled by firewall redirectors, etc. Linux has full support for this (FTP, IRC, etc.) capabilty but many routers do NOT (NetGear DOES).



iptables

In linux operating system, the firewalling is taken care of using netfilter. Which is a kernel module that decides what packets are allowed to come in or to go outside.

iptables are just the interface to netfilter. The two might often be thought of as the same thing. A better perspective would be to think of it as a back end and a front end.

To cover the fundamentals, firewalling is the idea of deciding which packets are allowed to go in/out of the system.

To decide which port is allowed to communicate to the outside world (or even on the localhost) is the firewall's responsibility. You would command it to either accept, reject or drop a packet.



Why use iptables?

- It's a full feature tool providing you with everything you need
- You get more flexibility regarding the things you want to with a packet
- It is more powerful feature than **firewalld** and **ufw** which are used commonly for filtering the packets in Linux Box.
- **firewalld** and **ufw** only allow you to accept or reject a packet.
- But there are lot more you can do with a packet. NAT, logging and forwarding are just a few to name.

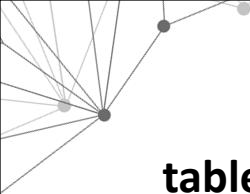
A decorative graphic in the top-left corner consisting of a network of nodes and connecting lines, resembling a web or a neural network structure.

iptables architecture

iptables consists of different components which are discussed below:

chains: There are 5 chains in iptables and each is responsible for a specific task.

- **Prerouting:** this chain decides what happens to a packet as soon as it arrives at the network interface. We have different options such as altering the packet, dropping a packet, or doing nothing at all.
- **Input:** This is one of the popular chains, If you want to open/block a port, this is where you'd do it.
- **Forward:** This chain is responsible for packet forwarding. Which is what the name suggests.
- **Output:** This chain is the one responsible for all your web browsing among many others. You can't send a single packet without this chain allowing it. You have a lot of options whether you want to allow a port to communicate or not.
- **Postrouting:** This chain is where packets leave their trace last, before leaving our computer. This is used for routing among many other tasks just to make sure the packets are treated the way we want them to.

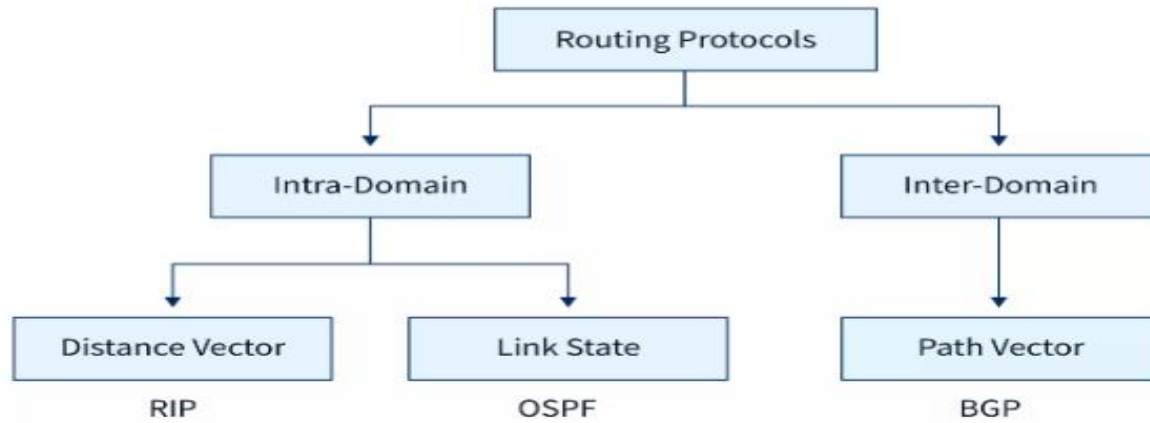


tables: Again, different tables are responsible for different tasks.

- **Filter:** This is the table most used on a daily basis. Which is why it's the default table. In this table you would decide whether a packet is allowed in/out your computer. If you want to block a port to stop receiving anything, this is your stop.
- **NAT:** This table is the second most popular table and is responsible for creating new connection.
- **Mangle:** For specialized packets only. This table is for changing something inside the packet either before coming in or leaving out.
- **Raw:** This table is dealing with the raw packet as the name suggests. Mainly this is for tracking the connection state.
- **Security:** It is responsible for securing your computer after the filter table.

Routing

Routing is the process of choosing a path for transferring data from a source to a destination. Routing is performed using devices called routers. In order to send the packet by determining the best route from one network to another, routing is carried out at the network layer. The network layer primarily makes sure that each packet arrives at its intended destination from the point of origin.



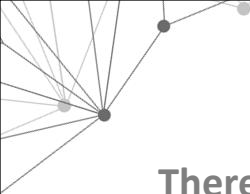


Interdomain Routing

Interdomain Routing is the protocol in which the routing algorithm works both within and between domains. Domains must be connected in some way, for hosts inside one domain to exchange data with hosts in other domains. This connection within domains is governed by the interdomain routing protocols. This is often done using the Border Gateway Protocol (BGP). It is used in Path Vector Routing using which interdomain routing is performed.

Intradomain Routing

Intradomain Routing is the routing protocol that operates only within a domain. In other words, intradomain routing protocols are used to route packets within a specific domain, such as within an institutional network for e-mail or web browsing. Unlike interdomain routing protocols, it doesn't communicate with other domains.



There are two types of protocols used for intradomain routing:

Distance Vector Routing (uses Routing Information Protocol or RIP) In distance vector routing, each node in a domain stores information about its neighboring nodes. The information is stored in a table known as a routing table, which is maintained by each node in the domain. RIP is one of the earliest distance-vector routing protocols, and it uses hop count as a routing statistic. By placing a cap on the maximum number of hops that may be taken between a source and a destination, RIP avoids routing loops.

Link State Routing (uses Open Shortest Path First or OSPF) In link state routing, each node in a domain stores information about all the other nodes in the domain, in other words, the routing table of each node stores information about the entire topology of the domain. Since each node has all the information about the domain at its disposal, Dijkstra's algorithm is used to calculate the best routing path. This is possible due to OSPF, and this is also its advantage.



Basis	Interdomain Routing	Intradomain Routing
Definition	The interdomain routing algorithms are used for routing within as well as with other domains.	The intradomain routing algorithms are used for routing within domains.
Router information	It requires information about the routers in the current domain as well as other domains.	It requires information only about the routers in the current domain.
Protocols	For interdomain routing, the protocols used are known as exterior-gateway protocols as they route traffic outside as well as inside a domain.	For intradomain routing, the protocols used are known as interior-gateway protocols as they route traffic within a domain.
Types	Interdomain routing is done using Path Vector Routing which uses the Border Gateway Protocol (BGP).	Intradomain Routing is of two types: Distance Vector Routing (uses Routing Information Protocol (RIP) and Link State Routing (uses Open Shortest Path First (OSPF).
Internet	The internet is assumed to be a collection of interconnected autonomous systems by the interdomain routing protocol.	The internet outside the autonomous system is ignored by intradomain routing protocols.

A decorative graphic in the top-left corner consisting of a network of nodes and connecting lines, resembling a partial graph or a network topology.

Unicast Routing

Unicast means the transmission from a single sender to a single receiver. It is a point-to-point communication between sender and receiver. There are various unicast protocols such as TCP, HTTP, etc.

- TCP is the most commonly used unicast protocol. It is a connection-oriented protocol that relies on acknowledgement from the receiver side.
- HTTP stands for HyperText Transfer Protocol. It is an object-oriented protocol for communication.



There are three major protocols for unicast routing:

- Distance Vector Routing
- Link State Routing
- Path-Vector Routing

Distance Vector Algorithm

- A router transmits its distance vector to each of its neighbors in a routing packet.
- Each router receives and saves the most recently received distance vector from each of its neighbors.
- A router recalculates its distance vector when:
 - It receives a distance vector from a neighbor containing different information than before.
 - It discovers that a link to a neighbor has gone down.



Link State Routing

- Link state routing is the second family of routing protocols.
- While distance-vector routers use a distributed algorithm to compute their routing tables, link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology.
- Based on this learned topology, each router is then able to compute its routing table by using the shortest path computation.

Path Vector Routing

- Path vector (PV) protocols, such as BGP, are used across domains aka autonomous systems.
- In a path vector protocol, a router does not just receive the distance vector for a particular destination from its neighbor; instead, a node receives the distance as well as path information (aka BGP path attributes)
- The node can use to calculate (via the BGP path selection process) how traffic is routed to the destination

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a mesh or star topology.

Multicast Routing

Multicast routing is a networking method for efficient distribution of one-to-many traffic. A multicast source, such as a live video conference, sends traffic in one stream to a multicast group. The multicast group contains receivers such as computers, devices, and IP phones.

Common uses include these technologies:

- Voice over IP (VOIP)
- Video on demand (VOD)
- Video conferencing
- IP television (IPTV)

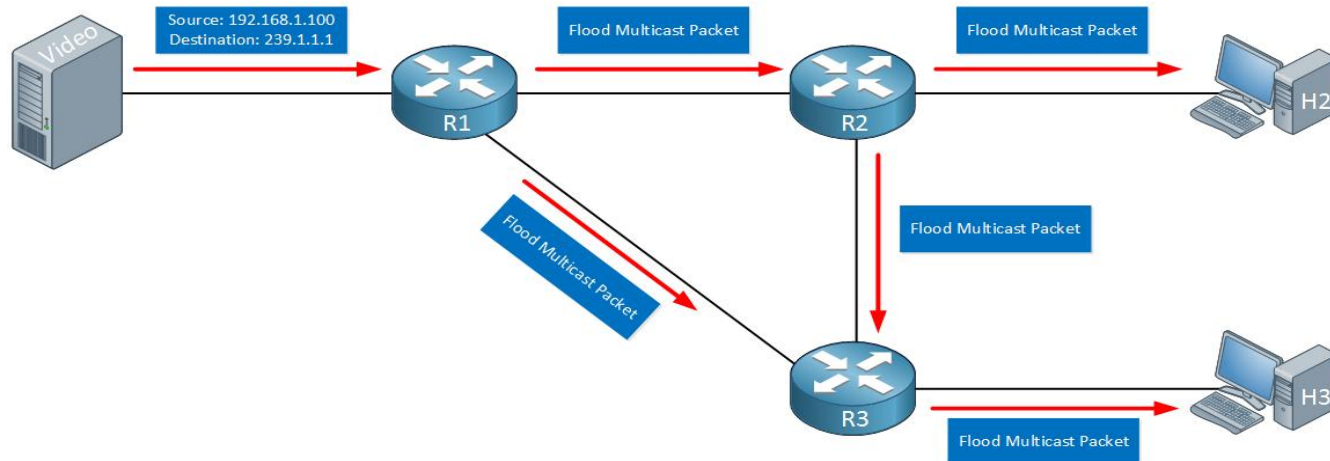
When you enable multicast routing on your Firebox, the Firebox acts as a local multicast router. It forwards multicast traffic from the source to receivers on your network.

There are two types of multicast routing protocols:

- Dense Mode
- Sparse Mode

Dense Mode

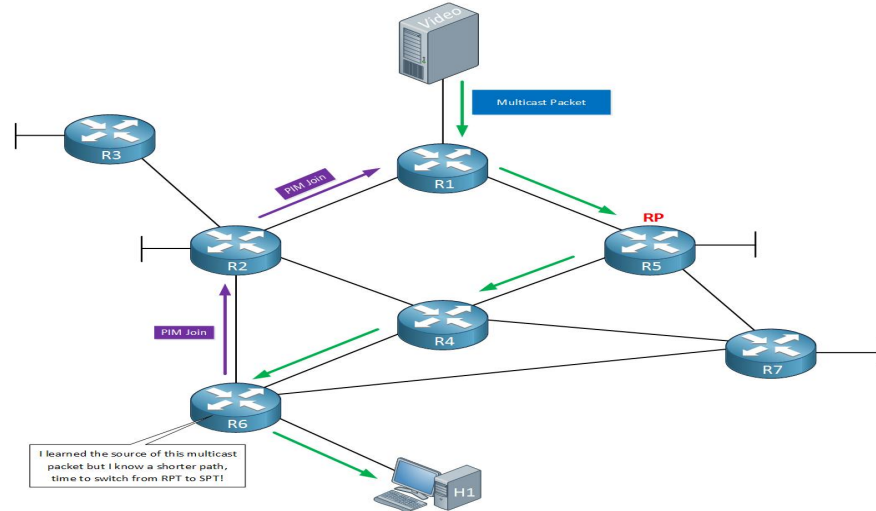
Dense mode multicast routing protocols are used for networks where most subnets in your network should receive the multicast traffic. When a router receives the multicast traffic, it will flood it on all of its interfaces except the interface where it received the multicast traffic on.



Sparse mode

As we can see that dense mode is very inefficient with its flooding of multicast traffic. When you only have a few receivers on your network then yes, you will be wasting a lot of bandwidth and resources on your routers.

The alternative is sparse mode which is far more efficient. Sparse mode multicast routing protocols only forward the multicast traffic when another router requests it. It's the complete opposite of dense mode:



A decorative graphic in the top-left corner of the slide, consisting of a network of nodes and edges. It features a central node with several lines radiating outwards to other nodes, some of which are further connected, forming a partial network structure.

Broadcast Routing

In broadcast routing, the network layer provides a service of delivering a packet sent from a source node to all other nodes in the network.

Broadcast Routing Algorithms

Perhaps the most straightforward way to accomplish broadcast communication is for the sending node to send a separate copy of the packet to each destination. Given N destination nodes, the source node simply makes N copies of the packet, addresses each copy to a different destination, and then transmits the N copies to the N destinations using unicast routing. This N -way unicast approach to broadcasting is simple—no new network-layer routing protocol, packet-duplication, or forwarding functionality is needed.

Duplicate creation/transmission

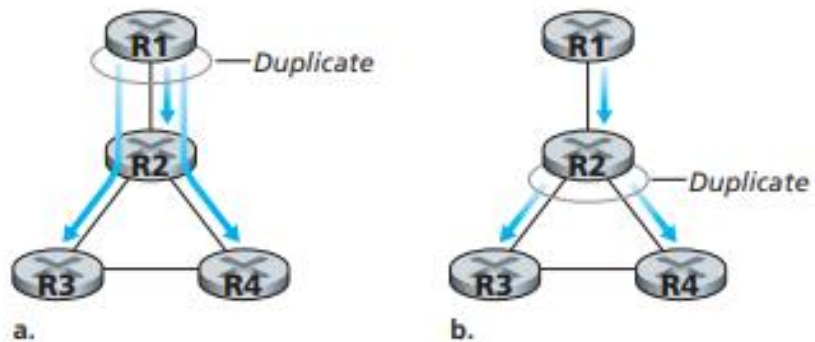


Figure 4.43 ♦ Source-duplication versus in-network duplication



Drawbacks to this Approach

- The first drawback is its inefficiency
- If the source node is connected to the rest of the network via a single link, then N separate copies of the (same) packet will traverse this single link.
- It would clearly be more efficient to send only a single copy of a packet over this first hop and then have the node at the other end of the first hop make and forward any additional needed copies.
- For example, in the figure above only a single copy of a packet traverses the R1-R2 link. That packet is then duplicated at R2, with a single copy being sent over links R2-R3 and R2-R4.
- Additional protocol mechanisms (such as a broadcast membership or destination-registration protocol) is required to know all the destination addresses which would add more overhead and, importantly, additional complexity to a protocol that had initially seemed quite simple.



Approaches to achieve broadcast routing efficiently

- Uncontrolled Flooding
 - The most obvious technique for achieving broadcast is a flooding approach in which the source node sends a copy of the packet to all of its neighbors. When a node receives a broadcast packet, it duplicates the packet and forwards it to all of its neighbors (except the neighbor from which it received the packet).

Drawback : **Broadcast Storm**

- Controlled Flooding
 - **Sequence-number-controlled flooding**
 - **Reverse path forwarding**
 - **Spanning-Tree Broadcast**

A decorative network diagram in the top-left corner showing several nodes (dots) connected by lines, representing a network topology.

Sequence-number-controlled flooding

In sequence-number-controlled flooding, a source node puts its address (or other unique identifier) as well as a broadcast sequence number into a broadcast packet, then sends the packet to all of its neighbors. Each node maintains a list of the source address and sequence number of each broadcast packet it has already received, duplicated, and forwarded. When a node receives a broadcast packet, it first checks whether the packet is in this list. If so, the packet is dropped; if not, packet is duplicated and forwarded to all the node's neighbors (except the node from which the packet has just been received).

Reverse path forwarding (RPF)

A second approach to controlled flooding is known as reverse path forwarding (RPF), also sometimes referred to as reverse path broadcast (RPB). The idea behind RPF is simple, yet elegant. When a router receives a broadcast packet with a given source address, it transmits the packet on all of its outgoing links (except the one on which it was received) only if the packet arrived on the link that is on its own shortest unicast path back to the source. Otherwise, the router simply discards the incoming packet without forwarding it on any of its outgoing links. Such a packet can be dropped because the router knows it either will receive or has already received a copy of this packet on the link that is on its own shortest path back to the sender.

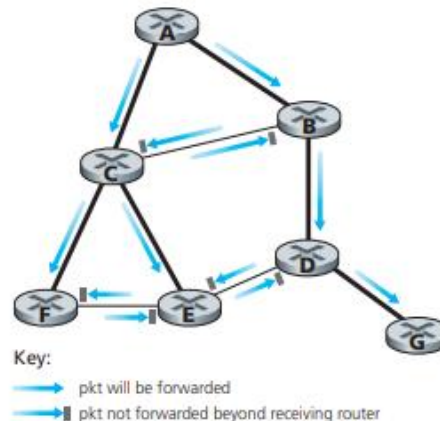


Figure 4.44 ♦ Reverse path forwarding

Spanning-Tree Broadcast

Another approach to providing broadcast is for the network nodes to first construct a spanning tree. When a source node wants to send a broadcast packet, it sends the packet out on all of the incident links that belong to the spanning tree. A node receiving a broadcast packet then forwards the packet to all its neighbors in the spanning tree (except the neighbor from which it received the packet). Not only does spanning tree eliminate redundant broadcast packets, but once in place, the spanning tree can be used by any node to begin a broadcast.

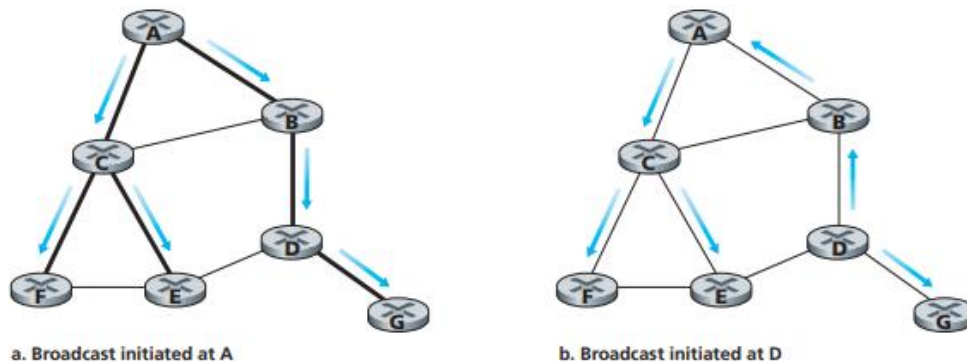


Figure 4.45 ♦ Broadcast along a spanning tree

2022

THANK YOU

