

Networking Concepts

By Jani Ahamed Habeeb Mohamed

Table of Contents

1. Introduction to Five Layer Networking Model	6
2. Network Devices	7
3. Physical Layer.....	8
4. Data Link Layer	9
5. Network Layer	10
6. Transport and Application Layer.....	17
7. Network Services	21
8. POTS, Dial Ups and Wireless.....	22
9. PKI.....	27
Scenario Overview:	28
Steps:.....	28
9.1.1.1 1. Configuration at the Office:	29
9.1.1.2 2. Home Router Configuration:	29
9.1.1.3 3. VPN Client Configuration:	29
9.1.1.4 4. VPN Connection Establishment:	29
9.1.1.5 5. Data Transfer:	29
9.1.1.6 6. Routing:	29
9.1.1.7 7. Office Local Network Access:	29
Key Points:	29
Components Involved:.....	30
Steps to Set Up a Tunnel:.....	30
Example using IPSec:	30
Key Considerations:	31
Components Involved:.....	31
Steps to Set Up a TLS VPN Tunnel:.....	32
Advantages of TLS VPN:	32
Considerations:	32
10. Layer 3 – ARP	35
Address Resolution Protocol (ARP) Message Format.....	36
Working of Address Resolution Protocol (ARP)	37
Address Resolution Protocol Explained with an example.....	38
11. Layer 3 – ICMP	41
11.1 ICMP Packet Format	41
11.2 ICMP in DDoS Attacks	42
Ping of Death Attack.....	42
ICMP Flood Attack.....	42
Smurf Attack	42
11.3 Types of ICMP Messages	42
Source Quench Message.....	44
Parameter Problem	45
Time Exceeded Message.....	45

Destination Un-reachable	46
Redirection Message	47
12. Layer 3 – VRRP	48
13. Layer 3 – BGP.....	49
14. Layer 3 – RIP	57
15. Layer 3 – OSPF.....	63
16. Layer 3 – IPsec VPN	70
16.1 IPsec components	70
16.2 IPsec encryption	70
16.3 IPsec modes: IPsec Tunnel vs. IPsec Transport	71
16.4 IPsec VPNs vs. SSL VPNs.....	71
17. Layer 3 – NAT	73
18. Layer 4 – TCP	76
18.1 Features	76
18.2 Header	76
18.3 Addressing.....	78
18.4 Connection Management	78
Establishment.....	79
Release.....	79
18.5 Bandwidth Management.....	80
18.6 Error Control &and Flow Control	80
18.7 Multiplexing.....	80
18.8 Congestion Control.....	81
18.9 Timer Management.....	81
Keep-alive timer:	81
Retransmission timer:	81
Persist timer:.....	81
Timed-Wait:	82
18.10 Crash Recovery	82
19. Layer 4 – UDP	83
19.1 Requirement of UDP.....	83
19.2 Features	83
19.3 UDP Header	83
19.4 UDP application.....	84
20. Layer 4 – TFTP	85
20.1 TFTP RRQ (Read Request) and TFTP WRQ (Write Request) messages	85
20.2 TFTP DATA packet size	85
20.3 TFTP port numbers	86
20.4 How TFTP (Trivial File Transfer Protocol) works	86

21.	Layer 7 – SOAP	89
22.	Layer 7 – SSDP.....	93
23.	Layer 7 – DHCP	94
24.	Layer 7 – DNS.....	97
25.	Layer 7 – HTTP	101
	Client.....	101
	Server	101
26.	Layer 7 – HTTPS - SSL?	106
26.1	Asymmetric Cryptography.....	107
26.2	Symmetric Cryptography	108
26.3	Data Transfer over SSL	108
	SSL Handshake.....	109
26.4	Actual Data Transfer.....	110
26.5	Public Key Infrastructure	111
27.	Layer 7 – SMTP	113
27.1	SMTP Protocol	114
27.2	Model of SMTP System	114
27.3	Components of SMTP.....	115
27.4	Working of SMTP	115
28.	Layer 7 – POP3.....	118
28.1	What is POP?	118
28.2	What is POP3?	118
	POP3 Ports.....	119
28.3	Working of POP3	119
28.4	Advantages of POP3.....	120
28.5	Disadvantages of POP3	120
28.6	Difference Between POP3 and IMAP	120
29.	Layer 7 – SSH	121
29.1	What is SSH Used for?	121
29.2	Establishing an SSH Connection	122
	Session Encryption Negotiation.....	122
	User Authentication.....	123
29.3	What is IMAP?	124
	Features of IMAP.....	124
29.4	Working of IMAP	124
29.5	Architecture of IMAP	124
29.6	Steps involve in IMAP Operation.....	125
30.	Layer 7 – NTP	127

31. Layer 7 – Telnet.....	129
31.1 Logging	129
31.2 Network Virtual Terminal(NVT).....	131
31.3 TELNET Commands	131
32. Layer 7 – FTP File Transfer Protocol (FTP) in Application Layer	134
32.1 What is File Transfer Protocol?	134
32.2 Types of FTP	134
32.3 How Does FTP Work?	135
32.4 Types of Connection in FTP	135
Control Connection	135
Data connection.....	136
32.5 FTP Session	136
32.6 FTP Clients	136
32.7 FTP Data Types	137
32.8 FTP Replies	137
32.9 Characteristics of FTP.....	137
32.10 FTP's Security Issues	137
32.11 Advantages of FTP	138
32.12 Disadvantages of FTP	138
32.13 Difference Between FTP and SFTP	138
33. Layer 7 – NFS	140
34. Glossary.....	143

1. Introduction to Five Layer Networking Model

Physical Layer: Bits, connectors and cables across devices with their signal patterns

Data Link Layer: Frames, Ethernet/Wifi protocol standards operating on MAC addresses. They define standards as to how these bit signals have to be interpreted in a common way for network devices to communicate. The Ethernet standard defines protocols to get data to nodes on the same network or link.

Network Layer: Packets/Datagrams, IP address, Allows different networks to communicate with each other through routers. Internetwork is a collection of networks connected through router (Internet). IP is the heart of the network.

Transport Layer: Segments, TCP/UDP, Ports, Determines which client and server programs are supposed to get the data.

Sessions Layer: Creates setup, control/tears down the connection between local and remote application. (Full/Half Duplex communication) and establishes procedure for checkpointing, suspending, resuming and terminating a session between two related streams of data such as audio and video in web conferencing.(implemented in places which use Remote Procedure Calls)

Presentation Layer: Translates/does certain things required by application layer to bundle messages before sending/while receiving. Data formatting (encoding/decoding), compression/decompression (ASN, XML, EBCDIC, ASCII)

Application Layer: Messages, HTTP/SMTP, different applications interact over web.

2. Network Devices

Cables :

Send signals between connected devices (point to point communication)

Copper : (multiple copper wires inside plastic insulator – changes voltage between two ranges to distinguish binary data and send it in cables – Cat5, Cat5e (reduce cross talk – i.e electrical pulse on one wire accidentally detected on another wire), Cat6 (most efficient to avoid cross talks, expensive and shorter range)– number of twists in pair of wires determine the transfer rate and usable links)

Fiber : Contains optical fibers that are Individual tiny tubes made out of glass about the width of human hair. Uses pulses of light to transmit binary information of underlying data. Used in places where there is lot of electro magnetic interference as it may impact signals sent in copper cables. Quickest, expensive, more fragile and bigger range of transmission without bigger loss.

Single network devices:

Primary devices used to connect computers in a local area network.

Hubs: Physical layer device that allows connection from many computers at once. Data sent from one system is received by every other system connected to the hub. It's the responsibility of the system to ignore unwanted messages. This causes a lot of unwanted noise and creates a collision domain (network segment where only one device can communicate at a time – if multiple devices communicate, there will be interference of signals (data collision), this will force the system to get into a quiet period and then try sending again)

Switches: Data Link layer device that can inspect the contents of the ethernet protocol frames and determines which system the data is intended for and only send data to that system. Eliminates collision domains resulting in higher throughput and fewer re-transmissions.

Multiple network devices:

Used to connect computers across different networks

Routers: Network layer device that knows how to forward data between independent networks. Router inspect IP data to determine where to send it to. Routers have an internal table which stores information about how to route data across the world. Most common routers are the one at home which takes data from home/office (home-office routers) and gives it to the ISP. The Internet service provider once has the data will use a sophisticated router(core/backbone routers) to send this data to the intended recipient. Core routers are connected to many other routers to determine the right path and use Border Gateway Protocol (to share data with each other router and to get to know the most optimal path to forward traffic)

Servers and Clients:

Something providing the data is the server entity and something requesting or getting that is the client entity.

3. Physical Layer

Send signals between connected devices (point to point communication)

Copper cables carries constant electrical charge and bits are sent though that by a process called Modulation.

Modulation is a way of varying this voltage of the charge moving across the cable.

Line coding – electrical charge in a certain state is 0 or 1 (10 billion 0s/1s every second)

Twisted Pair Cable Pairs of copper wires twisted together. (single conduit of information) and helps prevents interference with twisting. Cat6 – 4 twisted pairs (duplex communication – information can flow in both directions simultaneously across the cable where as simplex is unidirectional), say 2 pairs of twisted cables for one direction communication and the other 2 pairs for other way communication. When something is degrading, full duplex turns to half duplex which means there is still 2 way communication but only one entity can communicate at any point of time.

Twisted pairs cables use four pairs of color coded copper wires (one solid and one striped)

Uni-shielded Twisted Pair: common, least expensive for home/office, basic protection from Radio Frequency Interference and crosstalk.

Shielded Twisted Pair: used in environments where RFI is a big problem. Braided Aluminum/copper shielding underneath the outer jacket to shield the four pairs of copper cables.

Foiled Twisted Pair: used in RFI prone environments, uses a thin foil that wraps the pairs of copper bundle

Straight through cables are known as patch cables connect computers to hubs/switches.

A cable is straight through if color and stripe order are in same position on both ends

Computers and routers use: Pins 1 and 2 – orange wire for sending data Pins 3 and 6 – green wire for receiving data

Hubs and switches use: Pins 1 and 2 – green wire for sending data Pins 3 and 6 – orange wire for receiving data

Cross Over Cables: Used in older enterprise networks for direct connection.

Endpoint 1 of the Ethernet cable: Pins 1 & 2 - Green wires for sending data, Pins 3 & 6 - Orange wires for receiving data

Endpoint 2 of the Ethernet cable: Pins 1 & 2 - Orange wires for sending data Pins 3 & 6 - Green wires for receiving data

Network Ports and Patch Panels:

Twisted pairs are terminated with a plug (RJ45) that takes the individual internal wires and exposes them. RJ 45 cables are connected to RJ45 network ports on the devices. Link LED(will be lit when cable is properly connected between two powered on network devices) and Activity LED(will flash when the data transmission is happening) are present.

There might be network ports mounted on a wall. These ports will be connected to the network via cables that run through the walls and will eventually end in a patch panel (contains many network ports), its just a container for many end points. Cables then run from patch panels to switches or routers.

4. Data Link Layer

Ethernet – most common protocol used to send data across individual links. Provides an abstraction for the above layers from not caring about what is the kind of hardware used/physical layer.

Collision domain problem was solved by a technique called Carrier Sense Multiple Access with Collision Detection(CSMA/CD) – Used to determine when a communication channel is clear and is free to send data. Wait for a random interval when a collision is detected and then send.

Media Access Control Address (MAC) – globally unique identifier attached to a network interface. Its a 48 bit number represented by 6 groupings of 2 hexa decimal numbers

Octet: any number that can be represented by 8 bits

MAC address : 6 octets

First three octets : Organizationally unique identifier (OUI). These are assigned to HW manufacturers by IEEE.

Next three octets: Vendor assigned addresses

Unicast transmission is meant for one receiving address (Ethernet level – special bit in the destination MAC address)

Some quick examples of multicast MAC addresses: 01:00.CC:CC.DD:DD and 09:00.AA-AA.BB:BB

Some quick examples of unicast MAC addresses: 00:01:44:55:66:77 and 08:00:22:33:44:55

In an Ethernet MAC address, the least significant bit (LSB) of the first byte determines whether it is a unicast or multicast address. Specifically:

If the LSB is 0, it's a unicast address.

If the LSB is 1, it's a multicast address.

Let's examine the given MAC address: 0100.CCCC.DDDD

The LSB of the first byte (01) is 1.

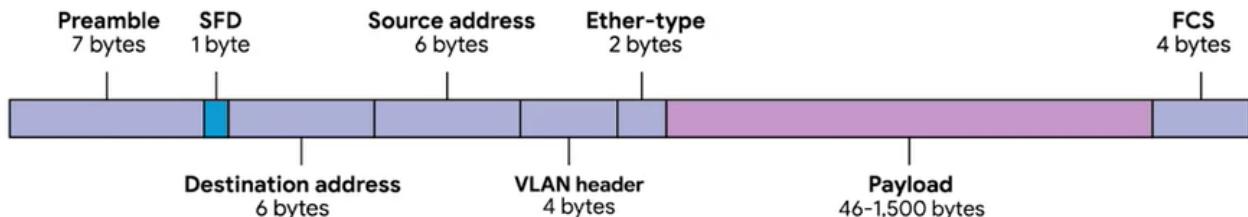
Therefore, based on the LSB, this MAC address is a multicast address.

Broadcast is all Fs FF:FF:FF:FF:FF:FF

One set of binary data being sent from one point to another is data packet.

A highly structured collection of information presented in a specific order is ethernet frame.

Dissect ethernet frame:



First 8 bytes (64 bits) – Preamble – split into two sections

First 7 bytes are bits if alternating 0s/1s used for internal clock synchronization for determining the speed of sending data or a buffer between frames and the last byte is the Start Frame Delimiter(SFD)

SFD will indicate the receiving device that the preamble is over and the actual data will follow

Destination MAC address 6 bytes : HW address of the intended recipient

Source MAC address 6 bytes: HW address of the sender.

Ether type : 2 bytes used to describe the protocol of the contents of the frame

Vlan Header : 4 bytes indicates this is a vlan frame, if a vlan header is present the ethertype follows it.

A virtual LAN is a technique that lets you have multiple logical LANs operating on the same physical equipment

Payload is actual data other than header 46 – 1500 bytes long

Frame check sequence : 4 bytes – checksum value for the entire frame. Checksum value is determined by Cyclical redundancy check against the frame. (used for data integrity)

A mathematical transformation that uses polynomial division to create a number that represents a larger set of data. It is an important concept for data integrity and is used all over computing, not just network transmissions

5. Network Layer

Dotted decimal format (12.34.56.78) – 32 bit 4 byte IP addresses.

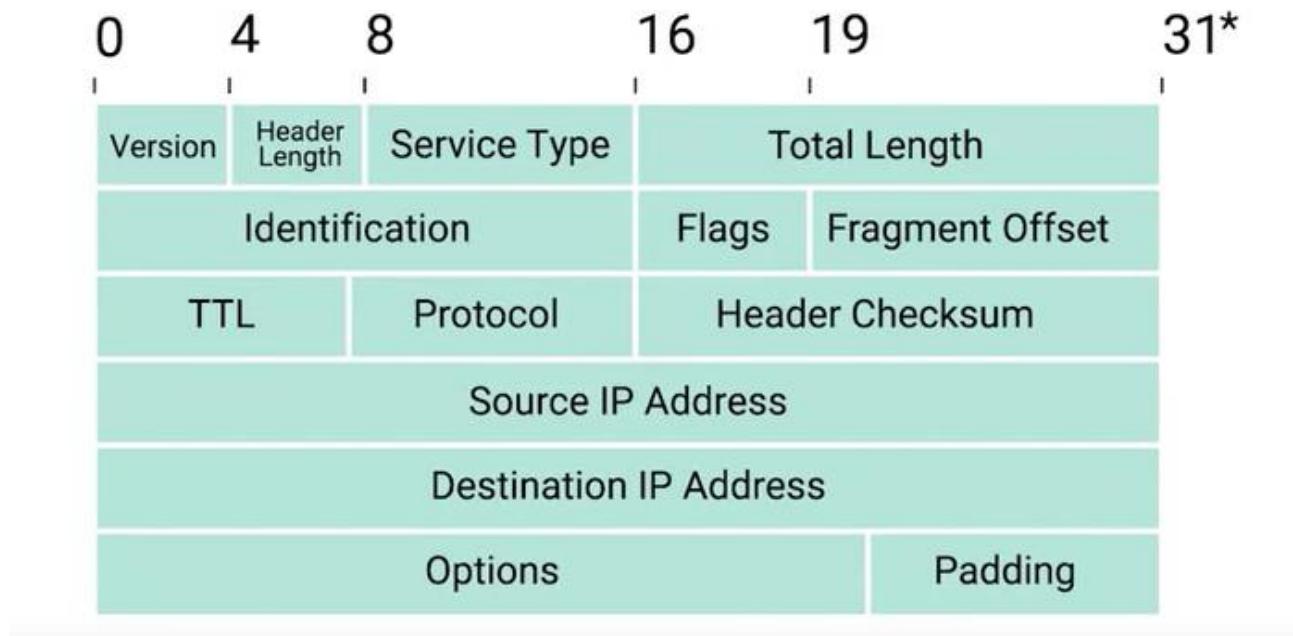
IP addresses belong to the networks not to the devices attached to those networks.

Dynamic IP addresses are assigned through Dynamic Host Configuration Protocol (DHCP)

Static IP addresses are manually configured (servers and network devices)

Data packets in network layer are termed as IP data grams (packets) – a highly structured series of fields that are strictly defined.

IP Datagram Header



Version (IPv4/IPv6) – first 4 bits

Header length (20 bytes (minimum length)– Ipv4) – 4 bits

Service Type field (8 bits – QoS related) – helps routers determine which IP datagram is more important

Total Length – 16 bits – total length of the IP datagram its attached to.

Identification – 16 bits (used to group messages together)

The maximum size of a single IP data gram is $2^{16} - 65535$

If the size of data is more than what could fit in a single IP datagram payload, then IP splits those and tags them with same identification number so that the receiver could identify they are all part of same IP datagram

Flag (3 bits) – indicates if a datagram is allowed to be fragmented or has already been fragmented.

Fragment offset (13 bits) : Helps receiver to identify and put all fragments of a single IP datagram together.

TTL : 8 bits indicates how many router hops an IP data gram can hop before being thrown away. The main purpose of having TTL is because when there is a misconfiguration in network that could lead to endless cycle for datagram to try reaching a destination, decrementing this value leading to 0 on the way will discard the packet.

Protocol (8 bits) – what kind of transport protocol is being used – tcp/udp

Header checksum : 16 bits checksum for the entire IP datagram header. Since TTL gets modified at each router, checksum also changes.

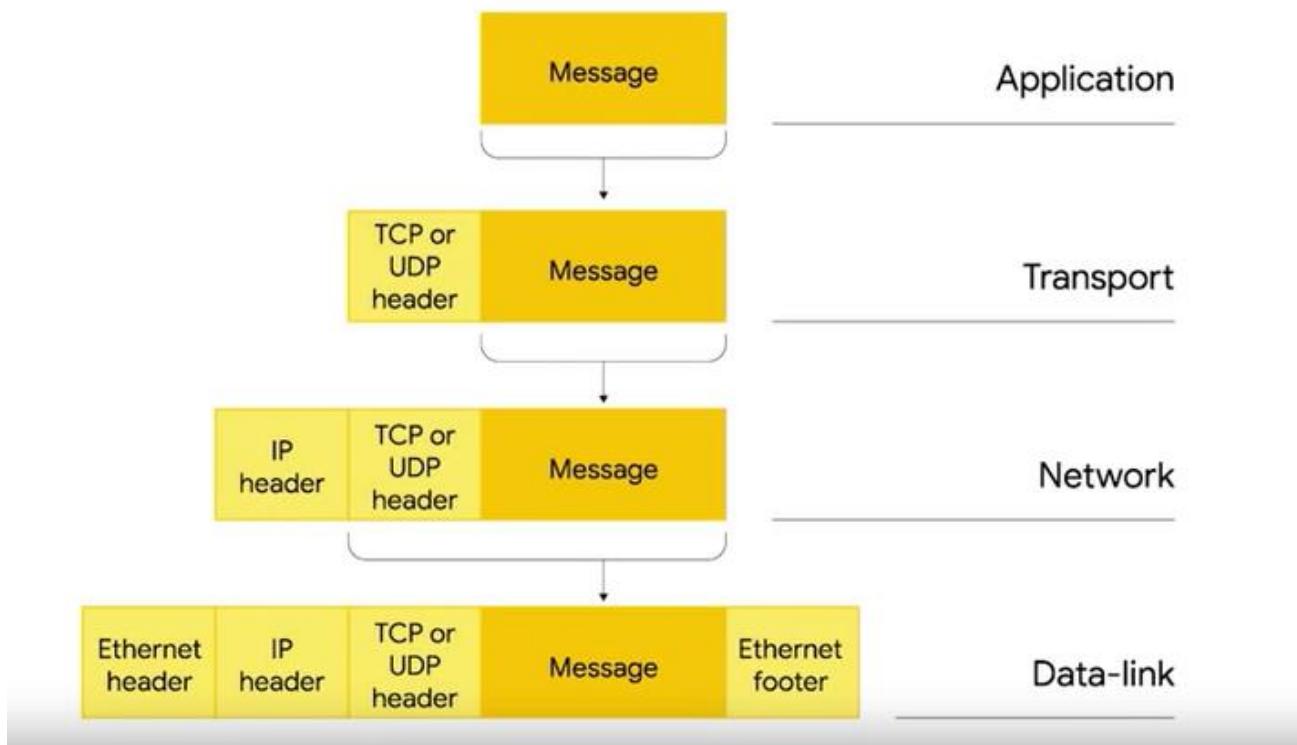
Src IP address – 4 bytes – 32 bits

Dst IP address – 4 bytes – 32 bits

Options field (20 bits) – set special characteristics for datagrams used for testing purpose

Padding field (12 bits) – variable just to accommodate to a series of 0s to ensure header is correct total size.

Encapsulation is putting all these IP datagrams inside the ethernet payload.



IP addresses have two sections : Network ID (first octet)/Host ID(2nd,3rd and 4th octet)
 Address class system : defining how global IP address space is split up.

Class A: Network ID (first octet)/Host ID(2nd,3rd and 4th octet)

Class B: Network ID (first, second octet)/Host ID(3rd and 4th octet)

Class C: Network ID (first, second and third octet)/Host ID(4th octet)

Class	Left-most bit	Starting IP address	Last IP address
A	Oxxx	0.0.0.0	127.255.255.255
B	10xx	128.0.0.0	191.255.255.255
C	110x	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255
E	1111	240.0.0.0	255.255.255.255

Class	First octet value	Subnet mask
A	0-127	8
B	128-191	16
C	192-223	24
D	224-239	-
E	240-255	-

Address resolution Protocol(ARP) – used to identify the MAC address tied to an IP address
 ARP table : list of IP and MAC address mapping

The process of taking a larger network and splitting it into smaller networks is subnetting



Picks up a portion of the host ID to form the subnet ID to break into smaller networks.

Core routers only care about network ID and send it to the Gateway router.

Gateway router uses additional information to send to the actual destination machine or the next router in path. Finally the host id is used to deliver the packet to the recipient.

Subnet masks are 32 bit numbers written out in four octets in decimal.

IP address	9	100	100	100
IP address (in binary)	0000 1001	0110 0100	0110 0100	0110 0100
Subnet mask (in binary)	1111 1111	1111 1111	1111 1111	0000 0000

255.255.255.0

First 8 bits are Class A network ID. Remaining are host IDs

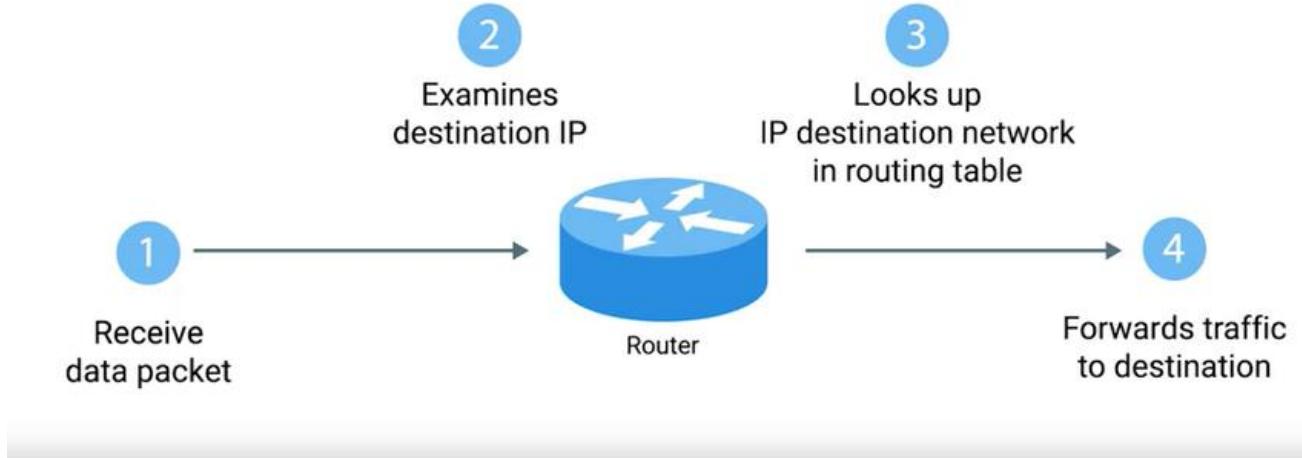
However since subnet mask is 255.255.255.0, the second and third octet are the subnet IDs and the last octet of the subnet mask, since its 0, the corresponding IP address part is the host ID.

Hosts can be assigned 1 – 254 as 0 is not used and 255 is from broadcast.

Router:

A router forwards traffic based on the destination IP address of that traffic.
 Router will atleast be connected to two networks to route traffic

Basic routing:



9.100.100.100

255 . 255 . 255 . 224

11111111 11111111 11111111 11100000

/27

Shorthand is to just tag ip with /number of 1 bits in subnet mask.

Classless Inter Domain Routing: (CIDR)

Demarcation point is where one network or system ends and another one begins

With subnets, network IDs were same size and portion of host id was used for subnet IDs.

But with CIDR, even the network IDs are usable

/24 network is 8 host bits. $2^8 = 256$

$256 - 2 = 254$

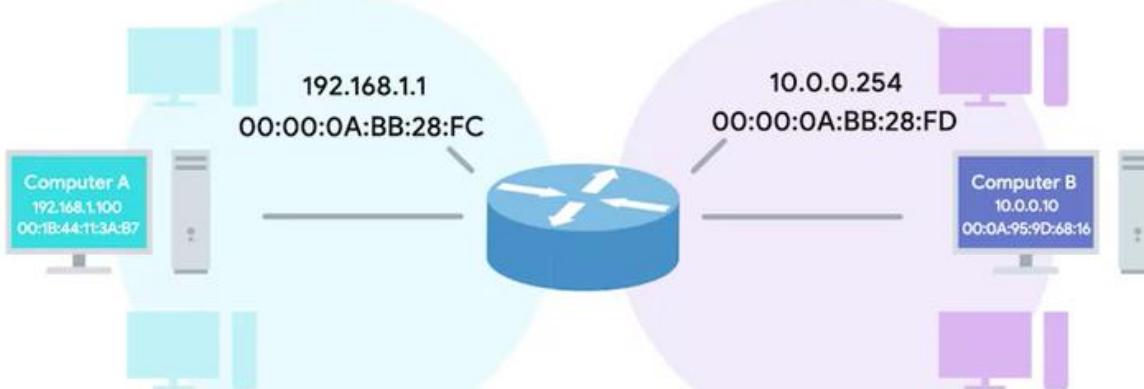
$254 + 254 = 508$

/23 network is 9 host bits. $2^9 = 512$

$512 - 2 = 510$

Network A
192.168.1.0/24

Network B
10.0.0.0/24



Packet from 192.168.1.100 to 10.0.0.10

So, Src: IP computer A, Dst IP: Computer B, Src MAC: Computer A, Dst MAC: Gateway

Computer A knows this IP is not local, so it sends it to the MAC address of the Gateway

Router now receives on interface 1 and removes the ethernet layer inspects the IP header and sees the destination IP field and gets to know from its routing table that the network 10.x.x.x is reachable via interface 2.

Now it constructs a new ethernet header with src MAC as his and dest MAC as Computer B MAC and decrements TTL and recalculates checksum.

Router has routing table and ARP table

So L2 MAC (src changes on every hop) during routing, whereas IPs remain the same.

MAC table:

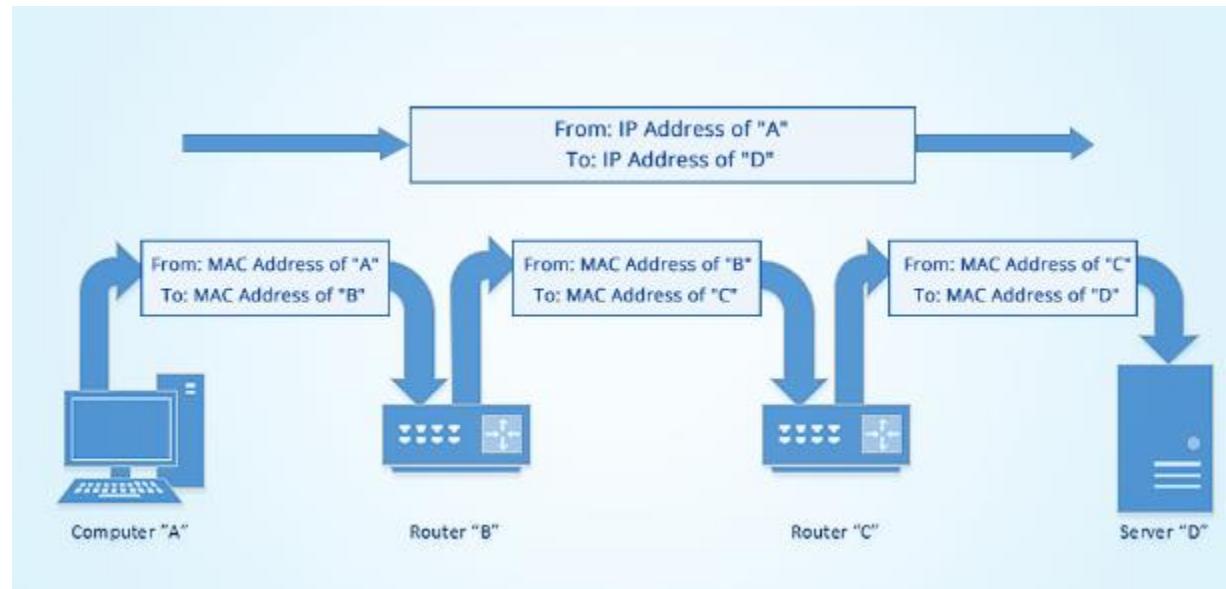
MAC Address	Address	Name	Interface	Flags
00:00:5e:00:01:0c	10.159.16.1	10.159.16.1	fxp0.0	none
fc:33:42:fe:dc:57	10.159.16.3	10.159.16.3	fxp0.0	none
9c:cc:83:8d:70:78	10.159.16.54	10.159.16.54	fxp0.0	none
00:a0:a5:7f:73:04	10.159.17.159	10.159.17.159	fxp0.0	none
00:a0:a5:9f:a0:c8	10.159.18.1	10.159.18.1	fxp0.0	none
44:aa:50:96:aa:19	30.65.0.2	30.65.0.2	fab0.0	permanent
30:b6:4f:8c:a7:07	30.66.0.1	30.66.0.1	fab1.0	permanent
fe:00:00:00:04:00	129.64.0.16	129.64.0.16	em0.0	none
fe:00:00:00:04:00	129.64.0.17	129.64.0.17	em0.0	none
aa:bb:cc:dd:ee:ff	192.168.1.1	192.168.1.1	em2.32768	none

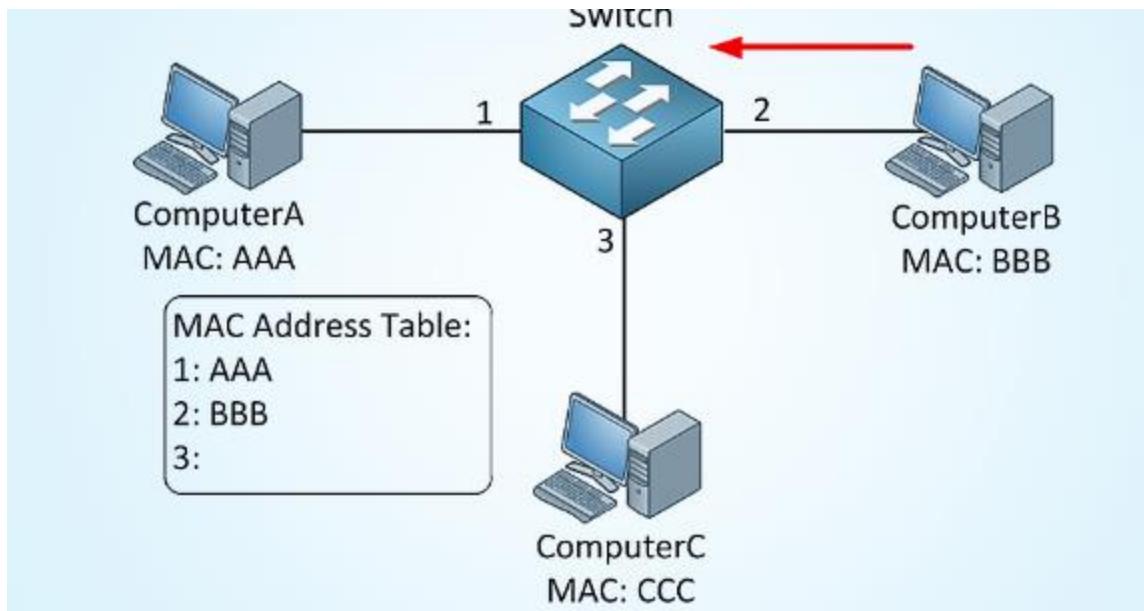
Routing table:

Basic Routing Table

Destination network	Next hop	Total hops	Interface
192.168.1.1/24	192.173.0.1	5	192.173.0.254
101.66.27.0/24	10.11.0.1	3	10.11.0.25
192.25.67.0/24	10.0.0.3	10	10.0.0.254

Nexthop is the ip address of the next router that should receive the data intended for the network under question and it is reachable via a certain interface and is certain hops away.





Routing protocols are used to learn about routes across the globe.

Interior (within an AS) and Exterior Gateway Protocol(between AS)

Interior : distance vector – every router has its own routing table where it maintains the distance or the number of hops for every other network and publishes it – no idea about the entire AS rather just basic information about neighboring router -> (RIP) and link state routing protocol (OSPF) – every interface of the router propagates all link related information to all other routers in the AS, then running Dijkstra's algorithm the optimized shortest path for a destination network is determined.

Exterior : Border Gateway protocol (between routers on the edge of AS)

IANA – Internet Assigned Numbers authority maintains assignment of ASN and IP address assignment.

Non routable Address Space: Ranges of IPs set aside for use by anyone that cannot be routed to.

In non routable address space computers talk to each other but no gateway router will try to reach this network.

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

Subnetting Host/Network

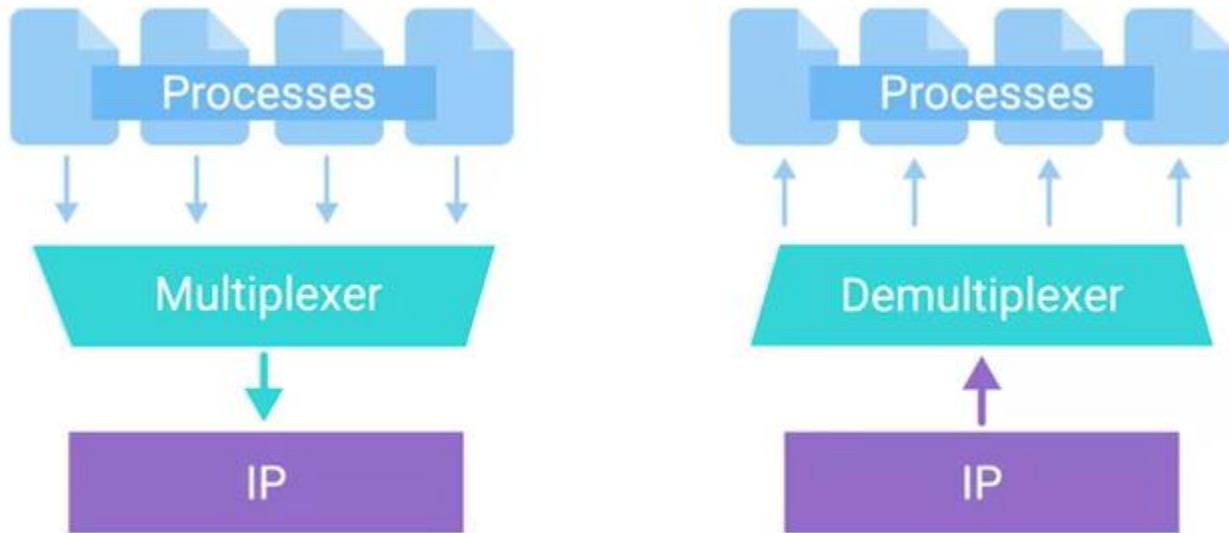
<https://www.computernetworkingnotes.com/ccna-study-guide/subnetting-tricks-subnetting-made-easy-with-examples.html>

6. Transport and Application Layer

Programs running on the system in the network should be able to talk to each other.

Transport layer allows traffic to be redirected to specific network applications

Application layer allows applications to communicate in a way they understand.



Multiplexing/Demultiplexing is done based on port numbers (16 bit number) that's used to redirect traffic to specific services in a computer.

10.1.1.100:80 – socket address/number

http : 80

ftp : 21

smtp : 25

printer : 9100

IP payload has TCP segment which is TCP header + data section.

Bit 0	Bit 15	Bit 16	Bit 31	
Source port (16)		Destination port (16)		
Sequence number (32)				
Acknowledgment number (32)				
Header Length (4)	empty (6)	Control flags (6)	Window (16)	20 Bytes
Checksum (16)		Urgent (16)		
Options (0 or 16 if any)		Padding		
Data payload (varies)				

Destination Port : Port of the service that the traffic is intended for

Source port or ephemeral ports : high numbered ports chosen from special section of ports (lot of outgoing connections separate)

Sequence number: 32 bit number used to keep track of where in a sequence of TCP segments this one is expected to be.

Acknowledgement number : number of the next expected segment

Data offset or Header length: 4 bit number that communicates how long the TCP header for this segment is

Empty : 6 bits

Control TCP flags : 6 bits

Window : 16 bits specifies the range of sequence numbers that might be sent before an acknowledgement is required

Urgent: is used in conjunction with tcp control flags to point out particular segments that might be more important than others.

Options: used for more complicated flow control protocols

Padding : makes sure payload starts at specified location.

Synchronization (SYN) – It is used in first step of connection establishment phase or 3-way handshake process between the two hosts. Only the first packet from sender as well as receiver should have this flag set. This is used for synchronizing sequence number i.e. to tell the other end which sequence number they should accept.

Acknowledgement (ACK) – It is used to acknowledge packets which are successful received by the host. The flag is set if the acknowledgement number field contains a valid acknowledgement number.

In given below diagram, the receiver sends an ACK = 1 as well as SYN = 1 in the second step of connection establishment to tell sender that it received its initial packet.

Finish (FIN) – It is used to request for connection termination i.e. when there is no more data from the sender, it requests for connection termination. This is the last packet sent by sender. It frees the reserved resources and gracefully terminate the connection.

Reset (RST) – It is used to terminate the connection if the RST sender feels something is wrong with the TCP connection or that the conversation should not exist. One of the sides of the TCP connection hasn't been able to properly recover from a series of missing malformed segments.

Urgent (URG) – It is used to indicate that the data contained in the packet should be prioritized and handled urgently by the receiver. This flag is used in combination with the Urgent Pointer field to identify the location of the urgent data in the packet.

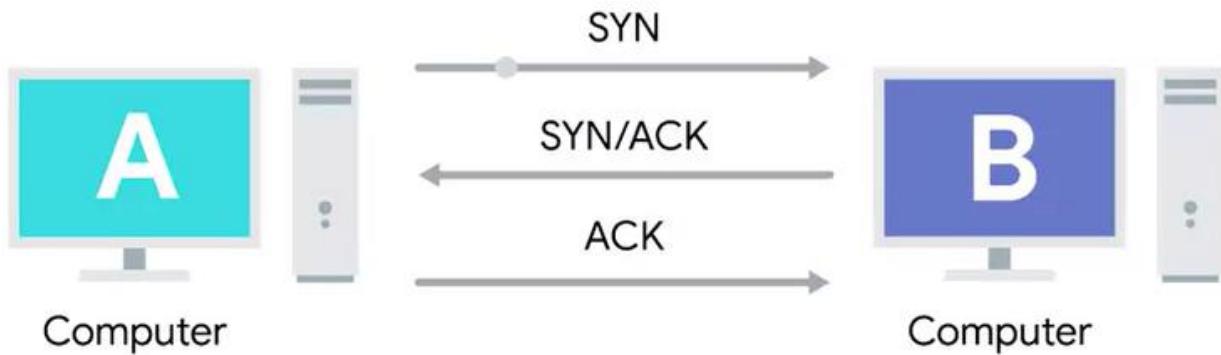
Push (PSH) – Transmitting device wants the receiver to push currently buffered data immediately to the application on the receiving side. It is used to request immediate data delivery to the receiving host, without waiting for additional data to be buffered on the sender's side. This flag is commonly used in applications such as real-time audio or video streaming.

Window (WND) – It is used to communicate the size of the receive window to the sender. The window size is the amount of data that the receiving host is capable of accepting at any given time. The sender should limit the amount of data it sends based on the size of the window advertised by the receiver.

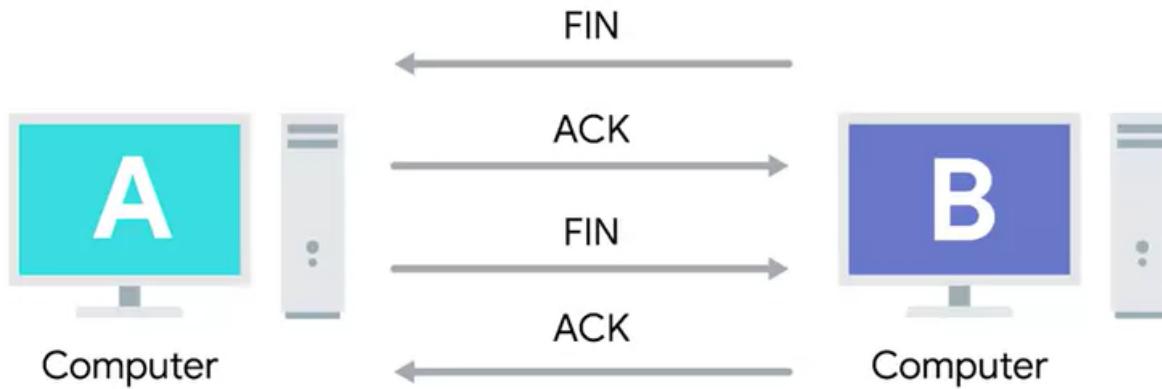
Checksum (CHK) – It is used to verify the integrity of the TCP segment during transmission. The checksum is computed over the entire segment, including the header and data fields, and is recalculated at each hop along the network path.

Sequence Number (SEQ) – It is a unique number assigned to each segment by the sender to identify the order in which packets should be received by the receiver. The sequence number is used in conjunction with the acknowledgement number to ensure reliable data transfer and to prevent duplicate packets.

The three-way handshake



The four-way handshake



Socket: The instantiation of an end point in a TCP connection.

Listen : A TCP socket is ready and listening for incoming connections (server)

SYN_SENT: A synchronization request has been sent but connection not established yet (client)

SYN_RECEIVED: A socket previously in Listen state has received a SYN request and has responded with SYN/ACK. (server)

Established: TCP connection is in working order and both sides are free to send data(client and server)

FIN_WAIT: A FIN has been sent but ACK has not been received

CLOSE_WAIT: The connection has been closed at the TCP layer but the application that opened that socket connection has not released hold on the socket yet.

CLOSED: The connection has been completely closed and no further communication is possible.

System Ports are identified as ports 1 through 1023. System ports are reserved for common applications like FTP (port 21) and Telnet over TLS/SSL (port 992). Many still are not assigned. Note: Modern operating systems do not use system ports for outbound traffic.

User Ports are identified as ports 1024 through 49151. Vendors register user ports for their specific server applications. The IANA has officially registered some but not all of them.

Ephemeral Ports (Dynamic or Private Ports) are identified as ports 49152 through 65535. Ephemeral ports are used as temporary ports for private transfers. Only clients use ephemeral ports

Firewall : device that blocks traffic that meet certain criteria

Apache Microsoft IIS and nginx are all web servers

Session layer is responsible for facilitating communication between actual applications and transport layer
Presentation layer makes sure unencapsulated application layer data is able to be understood by the application (encryption/compression).

7. Network Services

Virtual Private Networks:

Allows extension of a private or local network to hosts that might not work on the same local network

Tunnelling protocol

Remote client gets a virtual interface assigned with ip which is of the same address space of the network that the VPN is established to

VPNs work on the payload section of the Transport layer to carry an encrypted payload that carries an internal set of packets (network . transport and application layer)

Two factor authentication is used by VPNs

Proxy Service : A server that acts on behalf of a client in order to access another service

Anonymity, security, content filtering, increased performance.

Reverse Proxy: A service that might appear to be a single server to several clients but actually represents many servers living behind it.

8. POTS, Dial Ups and Wireless

Dial up connection uses POTS for data transfer by actually dialing a phone number. – MODEMS From PC digital to MODEM and then converted to ANALOG through telephone exchange.

Baud rate – how many bits can be transferred over a second in a phone line

Public switched telephone network

Anything that isn't dialup is broadband

T carrier technologies invented by AT&T to transmit multiple phone calls over single link

Digital Subscriber lines – Digital subscriber Lines Access Multiplexers (DSLAMs)

ADSL – Asymmetric – faster download and slower upload

SDSL – Symmetric – same speed upload/download

HDSL – High bit rate

Cable Broadband – shared bandwidth technology – managed by cable modem. Device that sits on edge of consumer network connect it to cable modem termination system

FTTX, FTTN – Fiber to the neighborhood and then COAX

FTTP – to the building

FTTH – to the home

FTTP – to the premises

Optical network terminator – converts data from protocol the fiber can understand to those that twice copper can understand.

Point to Point – Site to Site VPNs between Devices

IEEE 802.11 Wireless standard(wifi) – radio waves

Frequency band : A certain section of the radio spectrum that's been agreed upon to be used for certain communication(North America radio Frequency FM bands - 88 and 108MHz)

Wifi 2.4 and 5 Ghz bands

802.11b

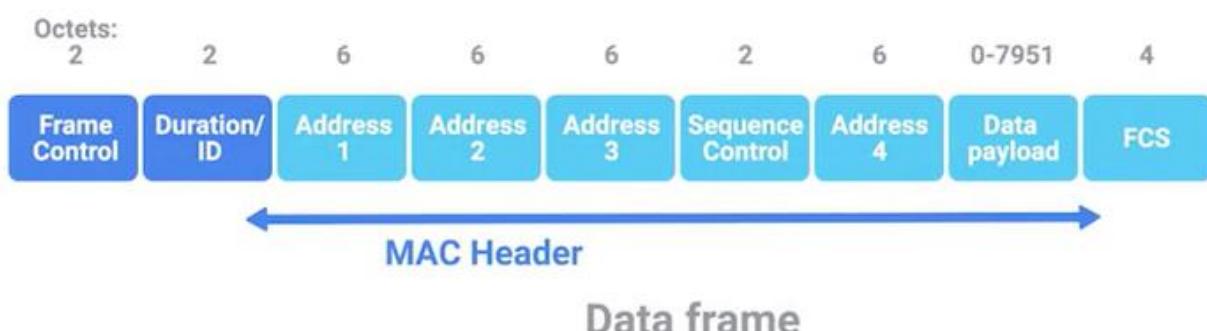
802.11a

802.11g

802.11n

802.1ac

802.11 – physical and data link layers



Frame control field

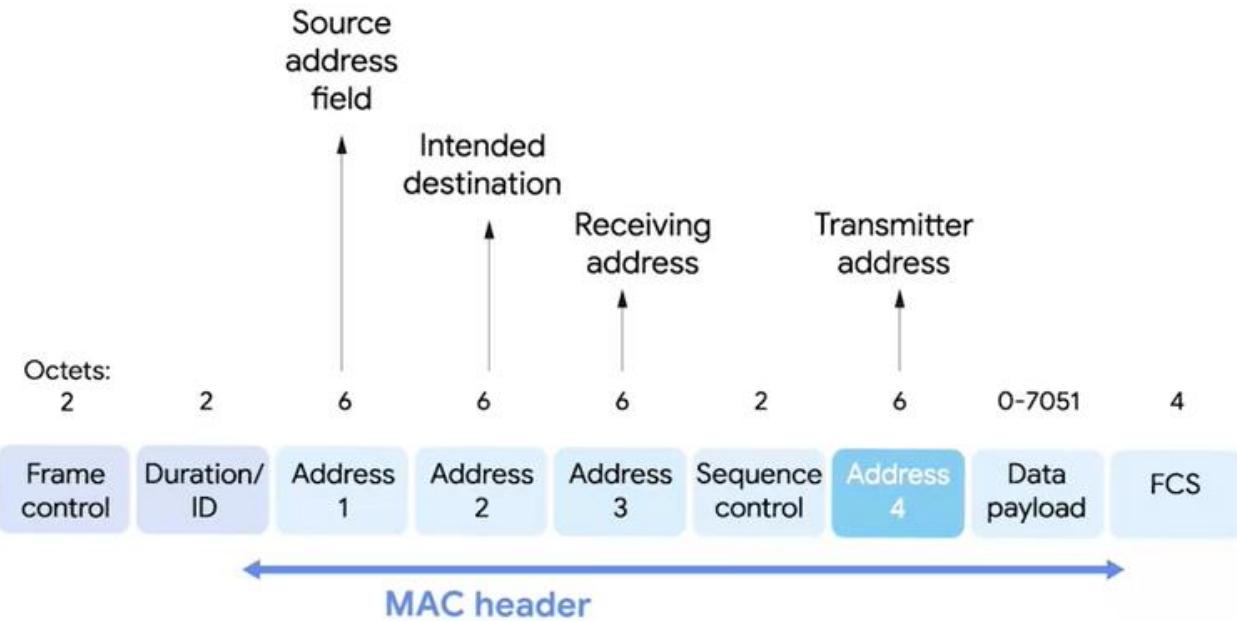
Is 16 bits long and contains a number of subfields that are used to describe how the frame itself should be processed

Duration field

It specifies how long the total frame is, so the receiver knows how long it should expect to have to listen to this transmission

Wireless access point

A device that bridges the wireless and wired portions of a network



Data frame

Adhoc networks where devices talk directly to each other

WLANS – where access points help talk

Mesh – hybrid of above

Channels are individual smaller sections of the overall frequency band used by wireless network

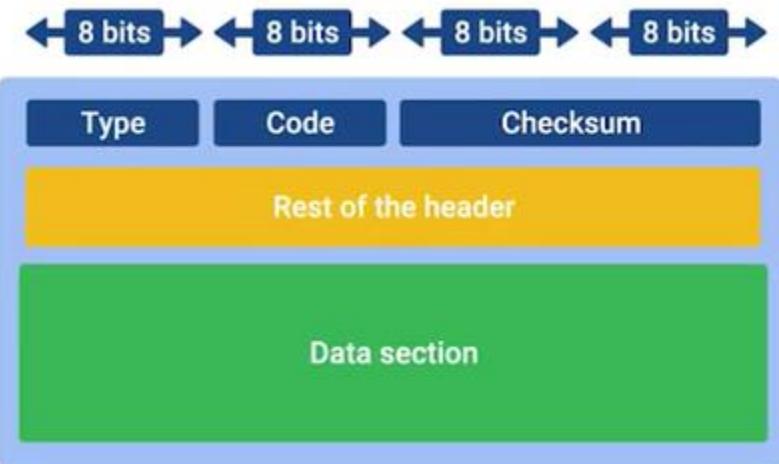
WEP (wired equivalent privacy) – low level privacy encryption

Wifi protected access

WPA2 – 256 bit key

MAC filtering

ICMP:



Traceroute send packets with increased ttl to find intermediate paths

Netcat nc (can send application data through keyboard) input is host and port

Nslookup for name resolution

Public name servers – anyone can use (level 3 – 4.2.2.1 to 4.2.2.6), 8.8.8.8 8.8.4.4 (google)

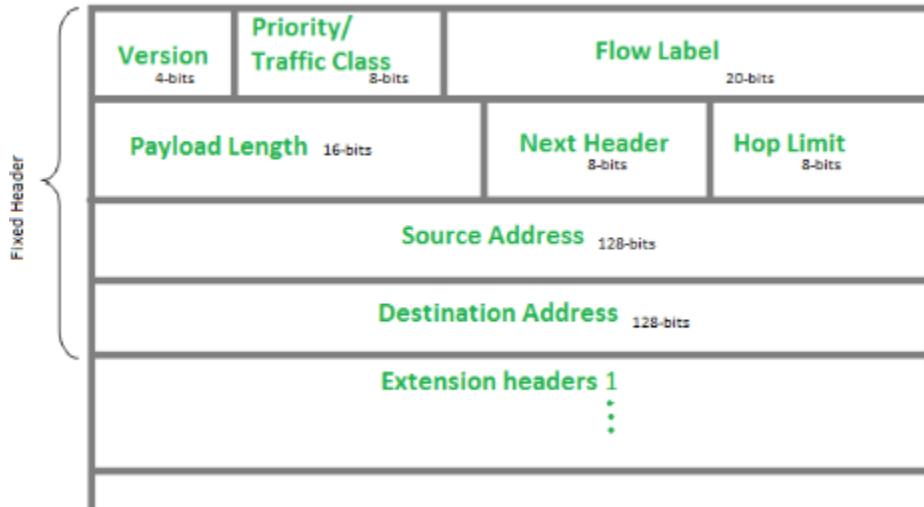
Original way numbered network addresses were correlated to words – host files

A single physical machine(host) could run many individual virtual instances called guests

Hypervisor : software that runs and manages the virtual machines while offering these guests a virtual

Operating platform that's indistinguishable from actual hardware

Public cloud – large cluster of machines run by another company
Private clouds – own premises physically hosted
IAAS - no worry about building own servers or network
PAAS – platform is provided to run their services
SAAS – software centrally managed and licensed for use
2001:0db8 – reserved ipv6 for documentation and education
FF00:: multicast
::1 loopback
FE80:: link local unicast addresses
64 network + 64 host = 128 bits



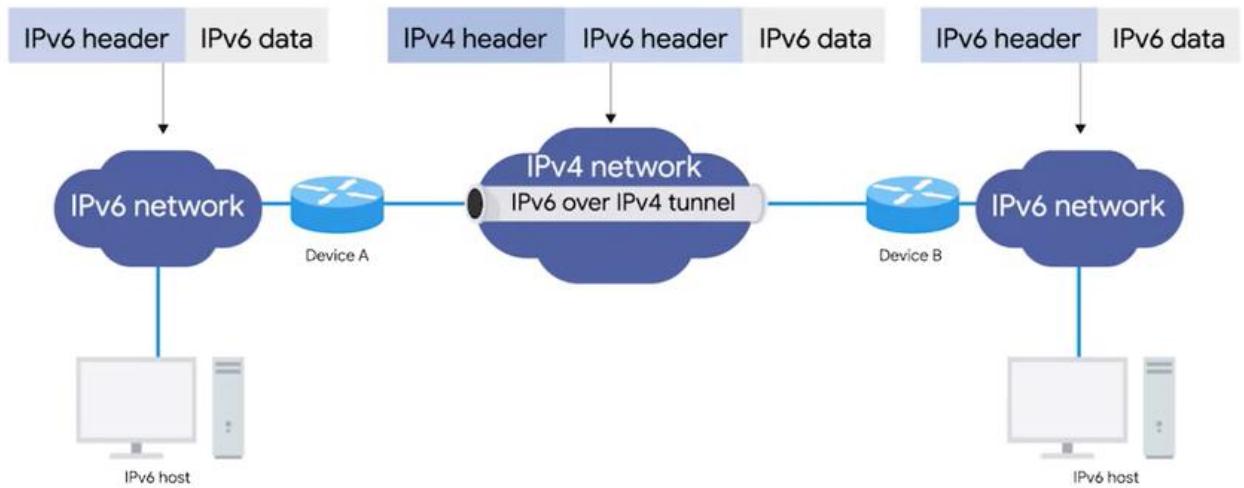
Version (4 bit) + 8 bit (traffic class) + flow label(20 bit) -> QOS + 16 bit payload length + 16 bit next header
8 bit hop limit ttl + src and dst ip + data payload

192.168.1.1 = 0:0:0:0:ffff:c0a8:0101

Mapped ipv4 address

IPv6 tunnels

Servers take incoming IPv6 traffic and encapsulate it within traditional IPv4 datagram



IPv6 tunnel broker

Companies that provide IPv6 tunneling endpoints for you, so you don't have to introduce additional equipment to your network

9. PKI

HTTP is stateless but not session less (maintains cookies to remember some states)

Server will place cookies in the browser cache and next time that information from cache is sent back to the server

First TCP established

Then HTTP method is sent – with url of the requested resource

TCP connected closed and recreated for every request

Multiplexing can be done with HTTP/2

On the URL -> there is protocol http/https

Uniform resource name – which has the domain/host.

Invisible connection port : 443 (https) or 80

Next is resource path

By default – index.htm or default.htm

?=Optional URL query- user id or other filtering actions

GET – 200 OK, 404 Not found 405 Not allowed 403 Forbidden (host with resource + authentication header or cookie)

POST – create a new resource(201 created 401 unauthorized 409 conflict 404 not found) – requests an id to track

PUT – Update an existing singleton resource based on ID (200 Ok 204 no content 404 not found 405 method not allowed)

PATCH – modify without replacing entirely based on ID

DELETE – remove a singleton resource based on ID

HEAD – just get the response header from the resource without touching it

OPTIONS – description of the communication option

TRACE – loopback of the request

Each method has a request response pair

1xx – information

2xx – success

3xx – redirection

4xx – client error

5xx – server error

SSL/TLS

Symmetric:

Rc4 – 2048

3des – 168

Blowfish – 448

AES – 256

Asymmetric:

RSA – 4096

Diffie hellman

Client sends – list of supported ciphers

Server selects strongest cipher and sends PKI certificate that contains servers public key and host name

Client generates and sends unique session key (encrypted with public key of server)

Webserver decrypts with private key.

Both sides know the symmetric session key

PKI components:

Certificate authority- issues, renews and revokes certificate

Registration authority – subordinate of CA – manages certificates

CRL – Certificate revocation list – Online certificate status protocol – OCSP – verification of certificate validity using a serial number

Certificate template – blueprint used when issuing certificate.

Certificate – signature of CA, expiry info and key

Certificate authorities issue certificates.

Certificates contain digital signature of the issuing CA which is signed by private key which can be validated with public key of that CA

PKI certificates – X.509 digital

Has version and serial number along with digital signature of CA, algorithm used, validity, usage details, URL or domain or email address, public key and private key

Wildcard certificate can be a single certificate that can match all sub domains under it

Stapling

Certificate owners can periodically check for their certificate validity, clients connecting to the server will receive OCSP status for that website

Public key pinning – through http header, trusting devices download a trusted copy of servers certificate which includes its public key

Lifecycle:

Certificate request - public and private key is generated

Private keys can be made available to trusted third parties

Certificate signing request is generated with the public key

CSR is sent to certificate authority for signing

Simple Certificate Enrollment Protocol

Certificate issuance – manual or automated

The certificate is stored in device trusted certificate store

Certificate Usage and Revocation

Certificate renewal – norm is 2 yrs

Certificate expiry

Encrypt for yourself – use private key of yours

Encrypt for others – public key of others

Explain how a computer at home can connect to office local network through VPN.. Explain in detail with private / public IP addresses

ChatGPT

ChatGPT

Sure, let's go through the steps of how a computer at home can connect to an office local network through a VPN, considering private and public IP addresses. This scenario typically involves a Remote Access VPN.

Scenario Overview:

- **Home Computer (Client):** The computer at home that wants to connect to the office network.
- **Home Router:** The router at the user's home that connects to the internet.
- **Internet:** The public network through which the home computer connects to the office network.
- **Office Router:** The router at the office that connects to the internet and the office local network.
- **Office Local Network:** The private network within the office.

Steps:

9.1.1.1 1. Configuration at the Office:

- **Public IP Address:** The office router has a public IP address assigned by the Internet Service Provider (ISP). This is the IP address visible on the internet.
- **VPN Server:** The office router is configured as a VPN server. It has a private IP address on its internal interface facing the office local network.
- **VPN Configuration:** The VPN server is configured with a VPN protocol (e.g., IPSec, SSL/TLS, or others) and authentication methods.

9.1.1.2 2. Home Router Configuration:

- **Public IP Address:** The home router also has a public IP address assigned by the ISP.
- **NAT (Network Address Translation):** The home router performs NAT, translating private IP addresses of devices within the home network to its public IP address when accessing the internet.

9.1.1.3 3. VPN Client Configuration:

- **VPN Client Software:** The home computer needs VPN client software. This software establishes a secure connection to the office VPN server.
- **VPN Configuration:** The user configures the VPN client with the public IP address of the office router, VPN protocol, and necessary authentication details.

9.1.1.4 4. VPN Connection Establishment:

- **Initiation:** The user initiates the VPN connection from the home computer.
- **Authentication:** The VPN client authenticates itself with the VPN server using pre-configured credentials or other authentication methods.
- **Secure Tunnel:** A secure and encrypted tunnel is established between the home computer and the office router over the internet.

9.1.1.5 5. Data Transfer:

- **Encrypted Data:** All data transmitted between the home computer and the office network is encrypted within the VPN tunnel, ensuring security.

9.1.1.6 6. Routing:

- **Routing Table Update:** Upon VPN connection, the routing tables are updated. The home computer knows to route traffic intended for the office local network through the VPN tunnel.

9.1.1.7 7. Office Local Network Access:

- **Access to Resources:** The home computer can now access resources on the office local network as if it were physically present in the office.

Key Points:

- **Public IP Addresses:** The routers at both the home and office have public IP addresses, but private IP addresses are used within the local networks.
- **Encryption:** The VPN tunnel ensures that data is encrypted, providing a secure connection.
- **Routing:** The routing tables are updated to direct traffic for the office local network through the VPN tunnel.

This setup allows a user working from home to securely connect to the office local network, access resources, and work as if physically present in the office.



You

how tunnel is setup?

ChatGPT

ChatGPT

Setting up a tunnel in the context of a VPN involves the creation of a secure, encrypted pathway through which data can be transmitted between two networks or devices. The process typically involves the configuration of various parameters, protocols, and security measures. Let's take a closer look at how a tunnel is set up:

Components Involved:

1. Tunnel Endpoints:

- Each end of the tunnel is referred to as a tunnel endpoint.
- In the case of a VPN, the tunnel endpoints are usually the routers or devices responsible for establishing the secure connection.

2. VPN Protocols:

- Various protocols can be used to implement VPN tunnels, such as IPSec (Internet Protocol Security), SSL/TLS (Secure Sockets Layer/Transport Layer Security), L2TP (Layer 2 Tunneling Protocol), and others.
- The choice of protocol depends on factors like security requirements, network architecture, and compatibility.

Steps to Set Up a Tunnel:

1. Configuration at Both Ends:

- Configure the settings on both ends of the tunnel, specifying details such as tunnel endpoints, encryption algorithms, authentication methods, and other parameters.

2. Authentication:

- Establish methods for authenticating the tunnel endpoints to ensure that only authorized devices can establish the tunnel.

3. Encryption Configuration:

- Specify the encryption algorithms and keys that will be used to secure the data transmitted through the tunnel.
- This ensures the confidentiality and integrity of the data.

4. Key Exchange:

- Establish a mechanism for securely exchanging encryption keys between the tunnel endpoints.
- Key exchange protocols, such as IKE (Internet Key Exchange), are commonly used for this purpose.

5. Tunnel Establishment:

- Initiate the process of tunnel establishment, during which the tunnel endpoints negotiate and agree on the parameters for secure communication.
- This involves the exchange of control messages to set up the tunnel.

6. Data Transmission:

- Once the tunnel is established, data can be transmitted through it securely.
- The data is encapsulated within the secure tunnel, protecting it from eavesdropping or tampering.

Example using IPSec:

1. Tunnel Configuration:

	<ul style="list-style-type: none"> Specify the tunnel endpoints' IP addresses. Define the encryption algorithm (e.g., AES), hash algorithm (e.g., SHA-256), and other parameters.
2. Authentication:	<ul style="list-style-type: none"> Choose an authentication method, such as pre-shared keys or digital certificates, to verify the identities of the tunnel endpoints.
3. Key Exchange:	<ul style="list-style-type: none"> Use a key exchange protocol like IKE to securely negotiate and exchange encryption keys between the tunnel endpoints.
4. Tunnel Establishment:	<ul style="list-style-type: none"> IKE initiates the negotiation process, and once agreement is reached, the tunnel is established.
5. Data Transmission:	<ul style="list-style-type: none"> Data is encapsulated within IPSec packets, ensuring secure transmission through the tunnel.

Key Considerations:

• Network Address Translation (NAT):	<ul style="list-style-type: none"> If either endpoint is behind a NAT device, additional considerations may be needed, such as NAT traversal mechanisms.
• Firewall Configuration:	<ul style="list-style-type: none"> Ensure that firewalls along the route allow the VPN traffic to pass through.
• Monitoring and Maintenance:	<ul style="list-style-type: none"> Regularly monitor the VPN tunnel for performance, and implement maintenance tasks such as updating keys.

Setting up a tunnel involves careful configuration and coordination between the tunnel endpoints to establish a secure and encrypted pathway for data transmission. The specific steps may vary based on the chosen VPN protocol and the devices involved.



You

explain tls vpn tunnel

ChatGPT

ChatGPT

Transport Layer Security (TLS) can be used to create a VPN tunnel, providing a secure and encrypted communication channel between two endpoints. This approach is often referred to as a TLS VPN or SSL VPN. TLS is commonly used to secure web traffic through HTTPS, but it can also be utilized to establish VPN connections. Here's an explanation of how a TLS VPN tunnel works:

Components Involved:

1. TLS Protocol:	<ul style="list-style-type: none"> TLS is a cryptographic protocol that ensures the confidentiality and integrity of data transmitted over a network. It operates at the transport layer of the OSI model, providing a secure communication channel.
2. VPN Server and Client:	

- The VPN server and client are the endpoints of the TLS VPN tunnel.
- The server typically resides in the target network (e.g., corporate network), while the client is a device connecting remotely.

Steps to Set Up a TLS VPN Tunnel:

1. Configuration of TLS Parameters:

- Specify TLS parameters such as the version of TLS (e.g., TLS 1.2, TLS 1.3), encryption algorithms (e.g., AES), and authentication methods (e.g., certificates).

2. Certificate Exchange:

- The VPN server and client exchange digital certificates to verify each other's identities.
- Certificates are issued by a trusted Certificate Authority (CA) and include public keys.

3. TLS Handshake:

- The TLS handshake process occurs between the server and client to negotiate encryption parameters and establish a secure connection.
- It involves several steps, including key exchange and mutual authentication.

4. Session Key Generation:

- During the TLS handshake, a session key is generated. This key is used for symmetric encryption, providing a secure and efficient way to encrypt and decrypt data during the session.

5. VPN Tunnel Establishment:

- Once the TLS handshake is successful, a secure TLS VPN tunnel is established between the server and client.
- All data transmitted through the tunnel is encrypted using the session key.

6. Data Transmission:

- Data is encapsulated within TLS/SSL packets and transmitted securely through the established tunnel.
- The TLS layer ensures that the data remains confidential and tamper-proof.

7. Authentication and Authorization:

- The server authenticates the client based on its digital certificate.
- Additional authorization checks may be performed to ensure that the client has the right to access specific resources.

Advantages of TLS VPN:

• Ease of Use:

- TLS VPNs are often easier to set up compared to other VPN technologies.
- TLS is widely supported by various devices and operating systems.

• Secure Communication:

- TLS provides strong encryption and ensures the confidentiality and integrity of data during transmission.

• Flexibility:

- TLS VPNs can be used for various applications, including remote access and site-to-site connections.

• Compatibility:

- TLS is commonly supported by web browsers and doesn't require special client software in many cases.

Considerations:

• Performance:

- While TLS provides strong security, the encryption and decryption processes can introduce some overhead, affecting performance.
- **Protocol Versions:**
 - Ensure that the chosen TLS version is up-to-date and secure, considering potential vulnerabilities in older versions.

A TLS VPN tunnel provides a secure and flexible way to establish encrypted connections over the internet, making it suitable for remote access scenarios or secure communication between networks. In a TLS VPN scenario, the TLS header and payload are encapsulated within the transport layer (commonly TCP) and the original IP and Ethernet headers are not changed. The TLS protocol operates at the application layer, securing the payload (application-layer data) within a TLS record. This TLS record, along with the original transport layer headers, is then transmitted over the network.

Here's a breakdown of the structure:

- 1. Original Ethernet, IP, and TCP Headers:**
 - The data transmission starts with the original Ethernet, IP, and TCP headers, which encapsulate the application-layer data.
- 2. TLS Record:**
 - The TLS Record encapsulates the application-layer data (TLS payload), the TLS header, and any necessary TLS Handshake messages.
 - The TLS Record includes metadata like the content type, version, and length.
- 3. Encrypted TLS Record:**
 - The TLS Record is encrypted, ensuring the confidentiality and integrity of the application-layer data.
- 4. Transport Layer Header (TCP/UDP):**
 - The encrypted TLS Record is treated as payload data and is encapsulated within the original transport layer header (commonly TCP, sometimes UDP in DTLS).
- 5. Original IP Header:**
 - The transport layer (TCP/UDP) header, including the encrypted TLS Record, is encapsulated within the original IP header.
- 6. Original Ethernet Header:**
 - The entire packet, including the original IP and transport layer headers with the encrypted TLS Record, is encapsulated within the original Ethernet header.

In summary, the TLS header and payload are part of the TLS Record, and the original IP and TCP headers are retained without modification. The TLS Record, along with the original transport layer headers, is transmitted over the network. The TLS encryption provides end-to-end security for the application-layer data, and the encapsulation within the transport layer ensures proper routing and delivery through the network.

10. Layer 3 – ARP

Address Resolution Protocol (ARP) is one of the major protocol of the TCP/IP protocol suite. The purpose of Address Resolution Protocol (ARP) is to resolve an IPv4 address (32 bit Logical Address) to the corresponding physical address (48 bit MAC Address). Network Applications at the Application Layer use IPv4 Address to communicate with another device. But at the Datalink layer, the addressing is MAC address (48 bit Physical Address), and this address is burned into the network card permanently. To understand this concept clearly, you need to visit the Ethernet frame format, which is the data packet generated at the Datalink layer. You can view your network card's hardware address by typing the command "ipconfig /all" at the command prompt (without double quotes using Windows Operating Systems).

In other words, there are two types of addresses used for IPv4 communication. The address used at layer 3 (Network layer) is IPv4 address (32 bit Logical Addresses) and the address used at layer 2 (Datalink layer) is MAC address (48 bit Physical Address). When a computer encapsulates an IPv4 datagram, it needs source IPv4 address, destination IPv4 address, source MAC address and destination MAC address. Source IPv4 address, destination IPv4 address, source MAC address are already available, because source IPv4 address and destination IPv4 addresses are available locally and the destination IPv4 address is where we are trying to communicate. At the layer 2 (Datalink layer) of the sending computer, if the computer want to encapsulate an IPv4 datagram inside an Ethernet frame, it needs destination MAC address. Address Resolution Protocol (ARP) is used to resolve layer 2 MAC Address of the receiver (destination MAC address).

Layer 3 (IPv4 datagram) - Source IPv4 address ✓
Layer 3 (IPv4 datagram) - Destination IPv4 address ✓
Layer 2 (Ethernet frame) - Source MAC address ✓
Layer 2 (Ethernet frame) - Destination MAC address X



©OmniSecu.com

The purpose of Address Resolution Protocol (ARP) is to find out the MAC address of a device in your Local Area Network (LAN), for the corresponding IPv4 address, which network application is trying to communicate.

[Address Resolution Protocol \(ARP\) Message Format](#)

Address Resolution Protocol (ARP) Message Format

Following are the fields in the Address Resolution Protocol (ARP) Message Format.

Hardware Type : Hardware Type field in the Address Resolution Protocol (ARP) Message specifies the type of hardware used for the local network transmitting the Address Resolution Protocol (ARP)

message. **Ethernet** is the common Hardware Type and its value for **Ethernet** is 1. The size of this field is 2 bytes.

Protocol Type : Each protocol is assigned a number used in this field. **IPv4** is 2048 (0x0800 in Hexadecimals).

Hardware Address Length : Hardware Address Length in the Address Resolution Protocol (ARP) Message is length in bytes of a **hardware (MAC) address**. **Ethernet MAC addresses** are 6 bytes long.

Protocol Address Length : Length in bytes of a **logical address (IPv4 Address)**. **IPv4 addresses** are 4 bytes long.

Opcode : Opcode field in the Address Resolution Protocol (ARP) Message specifies the nature of the ARP message. 1 for ARP request and 2 for ARP reply.

Sender Hardware Address : Layer 2 address (**MAC Address**) of the device sending the message.

Sender Protocol Address : The **protocol address (IPv4 address)** of the device sending the message

Target Hardware Address : Layer 2 (**MAC Address**) of the intended receiver. This field is ignored in requests.

Target Protocol Address : The **protocol address (IPv4 Address)** of the intended receiver.

Working of Address Resolution Protocol (ARP)

Step 1: When a source device wants to communicate with another device, source device checks its Address Resolution Protocol (ARP) cache to find if it already has a resolved **MAC Address** of the destination device. If it is there, it will use that **MAC Address** for communication. To view your local Address Resolution Protocol (ARP) cache, Open Command Prompt and type command "arp -a" (without double quotes using Windows Operating Systems).

Step 2: If ARP resolution is not there in local cache, the source machine will generate an Address Resolution Protocol (ARP) request message, it puts its own **data link layer address** as the Sender Hardware Address and its own **IPv4 Address** as the Sender Protocol Address. It fills the destination **IPv4 Address** as the Target Protocol Address. The Target Hardware Address will be left blank, since the machine is trying to find Target Hardware Address.

Step 3: The source **broadcasts** the Address Resolution Protocol (ARP) request message to the local network.

Step 4: The message is received by each device on the **LAN** since it is a **broadcast**. Each device compares the Target Protocol Address (**IPv4 Address**) of the machine to which the source is trying to communicate with its own Protocol Address (**IPv4 Address**). Those who do not match will drop the packet without any action.

Step 5: When the targeted device checks the Target Protocol Address, it will find a match and will generate an Address Resolution Protocol (ARP) reply message. It takes the Sender **Hardware Address** and the Sender **Protocol Address** fields from the Address Resolution Protocol (ARP) request message and uses these values for the Targeted Hardware Address and Targeted Protocol Address of the reply message.

Step 6: The destination device will update its Address Resolution Protocol (ARP) cache, since it need to contact the sender machine soon.

Step 7: Destination device send the Address Resolution Protocol (ARP) reply message and it will NOT be a **broadcast**, but a **unicast**.

Step 8: The source machine will process the Address Resolution Protocol (ARP) reply from destination, it store the Sender Hardware Address as the **layer 2 address** of the destination.

Step 9: The source machine will update its Address Resolution Protocol (ARP) cache with the Sender Hardware Address and Sender Protocol Address it received from the Address Resolution Protocol (ARP) reply message.

Address Resolution Protocol Explained with an example

Let us take an example of below topology.

I am sitting at 192.168.0.84. I want to ping to 192.168.0.122, to check the network connectivity. While running the ping command, we never specify the **MAC Address** of the destination device (192.168.0.122).

We know that ping command works using **ICMP**. **ICMP** is encapsulated inside **IP datagram** and **IP datagram** is encapsulated within **Ethernet Frame**. We need Source **IP Address** (my **IP Address**, 192.168.0.84), destination **IP Address** (192.168.0.122), Source **MAC Address** (my **MAC Address** 08:00:27:58:58:98) and destination **MAC Address** to make the **Ethernet Frame** for ICMP message. Source **IP Address**, destination **IP Address**, Source MAC Address are known at this instance, but the destination **MAC Address** unknown at this instance.

To assemble the **Ethernet Frame**, my device must have the destination **MAC Address** corresponding to the **IP Address** 192.168.0.122.

We need to resolve the destination **MAC Address** corresponding to the IP Address 192.168.0.122.

Step 1: Before resolving the destination **MAC Address** corresponding to the **IP Address** 192.168.0.122, using Address Resolution Protocol (ARP), source device checks its Address

Resolution Protocol (ARP) cache to find it already has a resolved **MAC Address** of the destination device. We can view the ARP cache by using the command arp -a

The ARP cache in my computer is empty. There are no entries in my ARP cache.

Step 2: Now my computer need to resolve the destination **MAC address** using ARP. My computer will prepare an ARP Request message and send it with a destination **MAC Address** as FF:FF:FF:FF:FF:FF (Broadcast **MAC Address**) to LAN Switch.

Step 3: Since the destination MAC Address is FF:FF:FF:FF:FF:FF (Broadcast MAC Address), the LAN Switch will flood it to all the connected ports and every device in the LAN will get a copy of it.

Following screen shot shows the Wireshark capture window of ARP Request message. You must compare the below screen shot with ARP message format image at the beginning of this lesson. We can see from the below screen shot that the destination **MAC Address** is FF:FF:FF:FF:FF:FF (Broadcast **MAC Address**), ARP opcode is 1 (for ARP Request), and the Target MAC Address is 00:00:00:00:00:00, which is unknown at this instance.

We can also see from the below screen shot that the Source **IP Address** is 192.168.0.84, destination **IP Address** is 192.168.0.122, Source **MAC Address** 08:00:27:58:58:98 and destination **MAC Address** is 00:00:00:00:00:00.

Step 4: The ARP Request message is received by each device on the **LAN** since it is a **broadcast**. Each device compare the Target Protocol Address (192.168.0.122), with its own IP Address. Those who do not match will drop the packet without any action.

Step 5: When the computer with the IP Address 192.168.0.122 receives the ARP Request, it must prepare an ARP Reply and send back to the computer who sent the ARP Request. ARP Reply will be a unicast, to save Network Resources.

Note that the ARP Reply has the Opcode filled as 2, which is used to identify it as a ARP Reply.

The "Sender MAC Address" field (which is marked below) in ARP Reply is the answer for ARP Request.

Now both the computers can update their ARP cache, so that the **MAC Address** to IP Address mappings can be used for a future communication. However, after a small period of time, ARP cache is flushed to avoid incorrect mappings (IP Address of any device can change at any time)

The ARP cache filled with a **MAC address** to IP Address mapping is shown below.

11. Layer 3 – ICMP

ICMP is used for reporting errors and management queries. It is a supporting protocol and is used by network devices like routers for sending error messages and operations information. For example, the requested service is not available or a host or router could not be reached.

Since the IP protocol lacks an error-reporting or error-correcting mechanism, information is communicated via a message. For instance, when a message is sent to its intended recipient, it may be intercepted along the route from the sender. The sender may believe that the communication has reached its destination if no one reports the problem. If a middleman reports the mistake

ICMP is the primary and important protocol of the IP suite, but ICMP isn't associated with any transport layer protocol ([TCP or UDP](#)) as it doesn't need to establish a connection with the destination device before sending any message as it is a connectionless protocol.

The working of ICMP is just contrasting with TCP, as TCP is a connection-oriented protocol whereas ICMP is a connectionless protocol. Whenever a connection is established before the message sending, both devices must be ready through a [TCP Handshake](#).

ICMP packets are transmitted in the form of datagrams that contain an IP header with ICMP data. ICMP datagram is similar to a packet, which is an independent data entity.

11.1 ICMP Packet Format

ICMP header comes after IPv4 and IPv6 packet header.

Type(8 bit)	Code(8 bit)	CheckSum(16 bit)
Extended Header(32 bit)		
Data/Payload(Variable Length)		

ICMPv4 Packet Format

In the ICMP packet format, the first 32 bits of the packet contain three fields:

Type (8-bit): The initial 8-bit of the packet is for message type, it provides a brief description of the message so that receiving network would know what kind of message it is receiving and how to respond to it. Some common message types are as follows:

- Type 0 – Echo reply
- Type 3 – Destination unreachable
- Type 5 – Redirect Message
- Type 8 – Echo Request
- Type 11 – Time Exceeded

- Type 12 – Parameter problem

Code (8-bit): Code is the next 8 bits of the ICMP packet format, this field carries some additional information about the error message and type.

Checksum (16-bit): Last 16 bits are for the checksum field in the ICMP packet header.

The checksum is used to check the number of bits of the complete message and enable the ICMP tool to ensure that complete data is delivered.

The next 32 bits of the ICMP Header are Extended Header which has the work of pointing out the problem in IP Message. Byte locations are identified by the pointer which causes the problem message and receiving device looks here for pointing to the problem.

The last part of the ICMP packet is Data or Payload of variable length. The bytes included in IPv4 are 576 bytes and in IPv6, 1280 bytes.

11.2 ICMP in DDoS Attacks

In Distributed DOS (DDoS) attacks, attackers provide so much extra traffic to the target, so that it cannot provide service to users. There are so many ways through which an attacker executes these attacks, which are described below.

Ping of Death Attack

Whenever an attacker sends a ping, whose size is greater than the maximum allowable size, oversized packets are broken into smaller parts. When the sender re-assembles it, the size exceeds the limit which causes a buffer overflow and makes the machine freeze. This is simply called a Ping of Death Attack. Newer devices have protection from this attack, but older devices did not have protection from this attack.

ICMP Flood Attack

Whenever the sender sends so many pings that the device on whom the target is done is unable to handle the echo request. This type of attack is called an ICMP Flood Attack. This attack is also called a ping flood attack. It stops the target computer's resources and causes a denial of service for the target computer.

Smurf Attack

Smurf Attack is a type of attack in which the attacker sends an ICMP packet with a spoofed source IP address. These type of attacks generally works on older devices like the ping of death attack.

11.3 Types of ICMP Messages

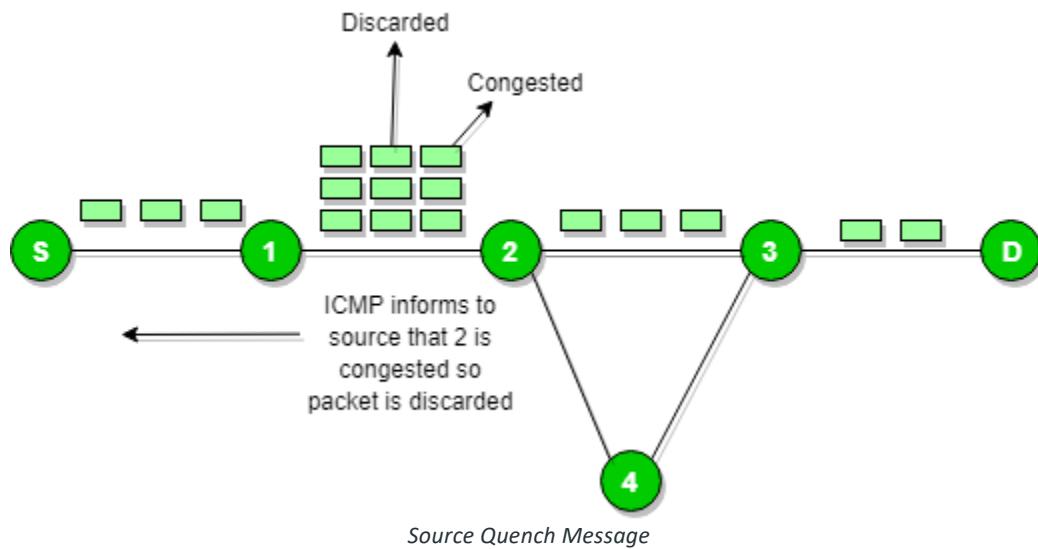
Type	Code	Description
0 – Echo Reply	0	Echo reply
3 – Destination Unreachable	0 1	Destination network unreachable Destination host unreachable

Type	Code	Description
5 – Redirect Message	2	Destination protocol unreachable
	3	Destination port unreachable
	4	Fragmentation is needed and the DF flag set
	5	Source route failed
	0	Redirect the datagram for the network
	1	Redirect datagram for the host
8 – Echo Request	2	Redirect the datagram for the Type of Service and Network
	3	Redirect datagram for the Service and Host
8 – Echo Request	0	Echo request
9 – Router Advertisement	0	Use to discover the addresses of operational routers
10 – Router Solicitation	0	
11 – Time Exceeded	0	Time to live exceeded in transit
	1	Fragment reassembly time exceeded.

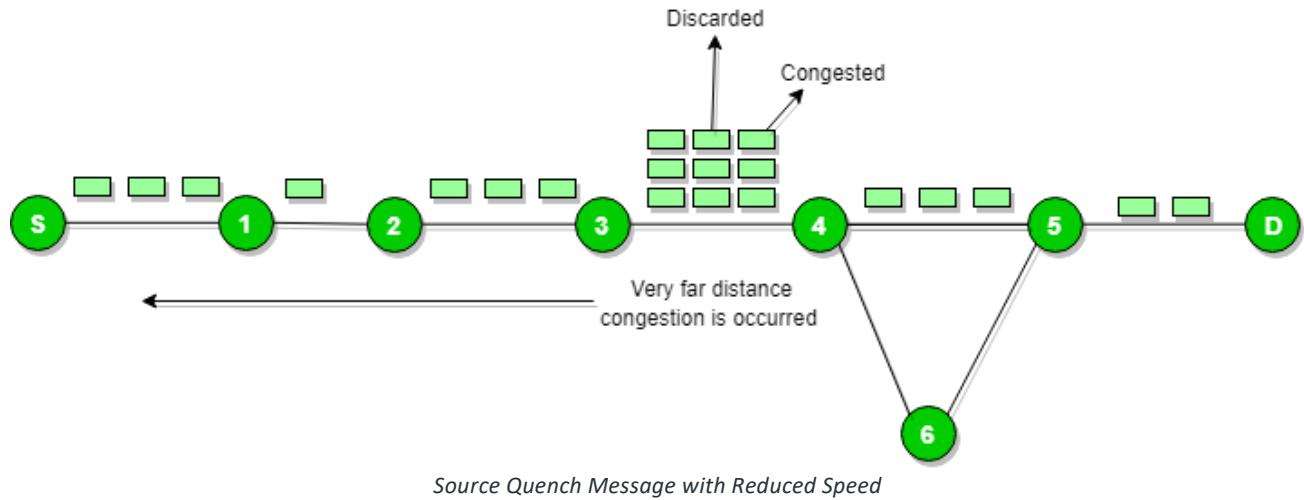
Type	Code	Description
12 – Parameter Problem	0	The pointer indicates an error.
	1	Missing required option
	2	Bad length
13 – Timestamp	0	Used for time synchronization
14 – Timestamp Reply	0	Reply to Timestamp message

Source Quench Message

A source quench message is a request to decrease the traffic rate for messages sent to the host destination) or we can say when receiving host detects that the rate of sending packets (traffic rate) to it is too fast it sends the source quench message to the source to slow the pace down so that no packet can be lost.



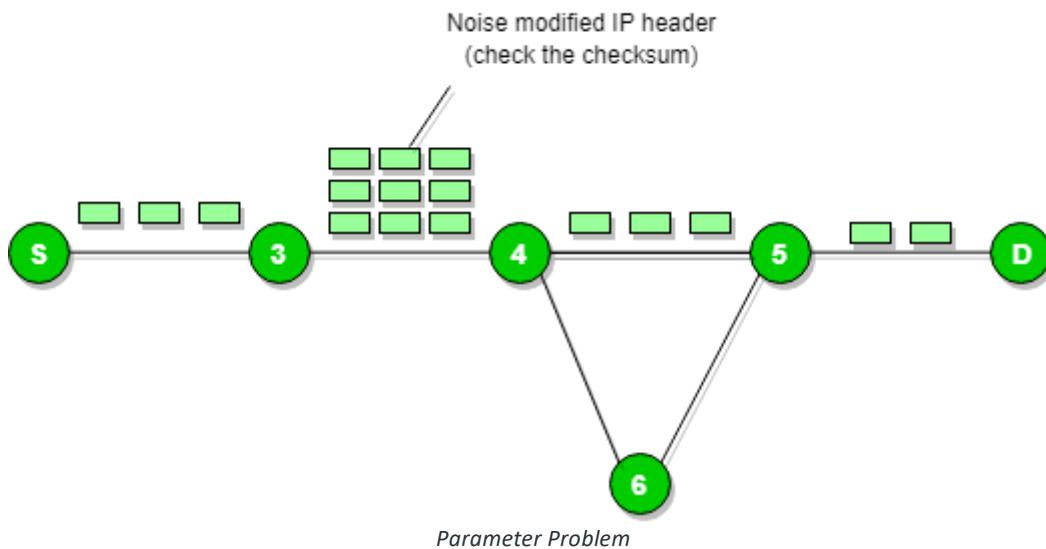
ICMP will take the source IP from the discarded packet and inform the source by sending a source quench message. The source will reduce the speed of transmission so that router will be free from congestion.



When the congestion router is far away from the source the ICMP will send a hop-by-hop source quench message so that every router will reduce the speed of transmission.

Parameter Problem

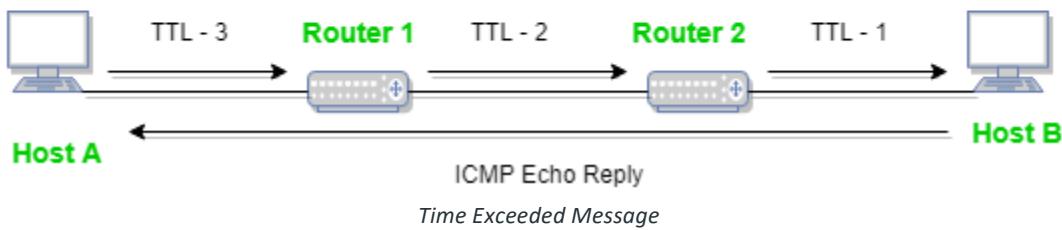
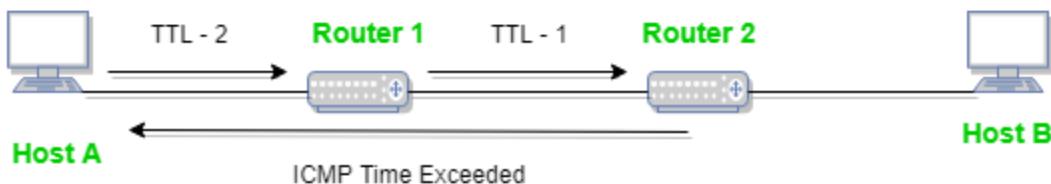
Whenever packets come to the router then the calculated header checksum should be equal to the received header checksum then only the packet is accepted by the router.



If there is a mismatch packet will be dropped by the router.

ICMP will take the source IP from the discarded packet and inform the source by sending a parameter problem message.

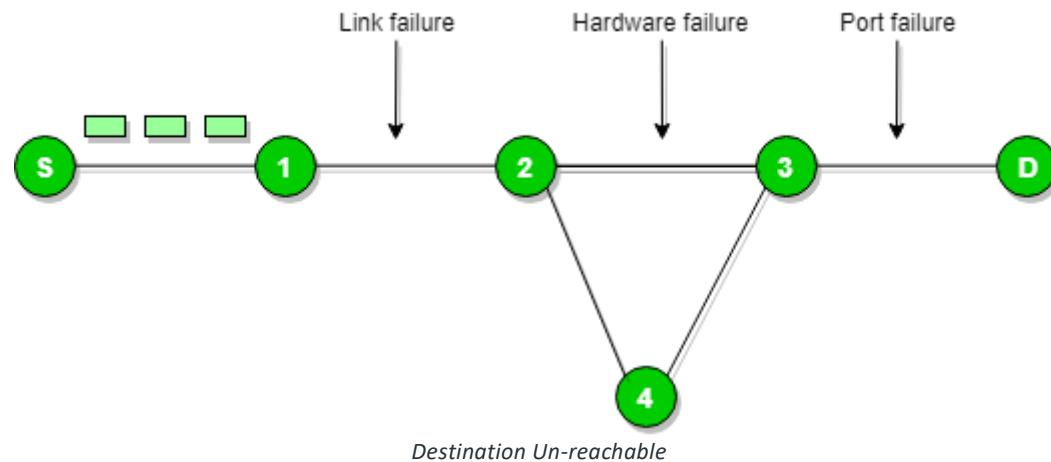
Time Exceeded Message



A notification with the subject line “Time Exceeded” is typically generated by routers or gateways. You need to know what an IP header is in a packet in order to comprehend this ICMP message in its entirety. The IP protocol structure is covered in great detail in the section on IP Protocol, which is freely available to our readers.

Destination Un-reachable

The destination is unreachable and is generated by the host or its inbound gateway to inform the client that the destination is unreachable for some reason.



There is no necessary condition that only the router gives the ICMP error message time the destination host sends an ICMP error message when any type of failure (link failure, hardware failure, port failure, etc) happens in the network.

Redirection Message

Redirect requests data packets are sent on an alternate route. The message informs a host to update its routing information (to send packets on an alternate route).

Example: If the host tries to send data through a router R1 and R1 sends data on a router R2 and there is a direct way from the host to R2. Then R1 will send a redirect message to inform the host that there is the best way to the destination directly through R2 available. The host then sends data packets for the destination directly to R2.

The router R2 will send the original datagram to the intended destination.

But if the datagram contains routing information then this message will not be sent even if a better route is available as redirects should only be sent by gateways and should not be sent by Internet hosts.

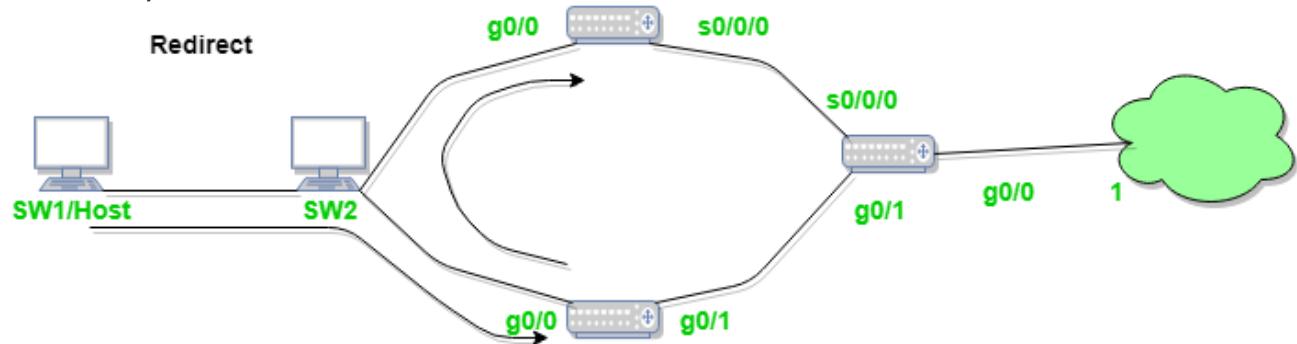


Figure - ICMP redirect Verification CCNP 2.0 100 - 101 (v - 71)

- ✓ ICMP Redirect
- ✓ ICMP Redirect for host
- ✓ ICMP Redirect for network
- ✓ How ICMP redirect work
- ✓ ICMP Redirect verification step by step

Redirection Message

Whenever a packet is forwarded in the wrong direction later it is re-directed in a current direction then ICMP will send a re-directed message.

12. Layer 3 – VRRP

VRRP is an open standard protocol, which is used to provide redundancy in a network. It is a network layer protocol (protocol number-112). The number of routers (group members) in a group acts as a virtual logical router which will be the default gateway of all the local hosts. If one router goes down, one of the other group members can take place for responsibility for forwarding the traffic.

Some important terms related to VRRP:

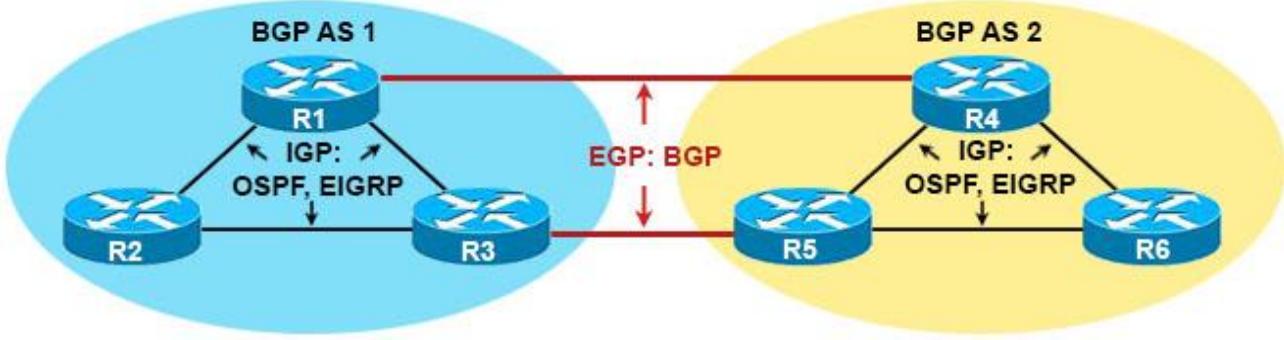
1. **virtual IP address:** An IP address is assigned as a virtual IP address from the local subnet which is configured as a default gateway for all the local hosts.
2. **Virtual MAC address:** A virtual MAC address is automatically generated by taking the last 8 bytes as the VRRP group number in hexadecimal. In VRRP, Mac address used is 0000.5e00.01xx. Here, xx is the VRRP group number in hexadecimal.
3. **Master router:** One of the VRRP group member is elected as master router which takes up the responsibility of forwarding the local traffic. The router is elected on the basis of priority. If some group member of VRRP group has higher priority than others then it will be elected as master router. If the priority is same(by default 100) then the router having the highest IP address will become the master router.
4. **Backup routers:** Only one of the VRRP group member will become the master router while others will be back up routers. In case the master router fails then one of the backup routers will become master router.
5. **Master advertisement timer:** The master router multicast the keep-alive messages at 224.0.0.18 in every 1 second.
6. **Master dead timer:** The time in which the backup router will take up the responsibilities of Master router if the master advertisement message is not received. It is, by default, 3.69 seconds.
7. **Preempt:** It is a state in which one of the backup routers becomes the master router (when the master router goes down). Also, when the master router comes up again, it will become the master router as its priority is still higher.
8. **Object tracking:** The VRRP object tracking provides a way in which the best router will become the master router. The current master router continuously polls the tracked object state (such as line protocol is up or down). Suppose if the master router line protocol is down then the priority of backup router will be increased dynamically i.e it will become the master router.
9. **Authentication:** VRRP supports 3 types of authentication – 1. No authentication 2. Plain text authentication 3. MD5 authentication

13. Layer 3 – BGP

Basic understanding about BGP

We really want to show you why we need BGP first but it is very difficult to explain without understanding a bit about BGP. So we will learn some basic knowledge about BGP first.

First we need to understand the difference between Interior Gateway Protocol and Exterior Gateway Protocol, which is shown below:



– **Interior Gateway Protocol (IGP)**: A routing protocol operating within an Autonomous System (AS) like OSPF, EIGRP... Usually routers running IGP are under the same administration (of a company, corporation, individual)

– **Exterior Gateway Protocol (EGP)**: A routing protocol operating between different AS. BGP is the only EGP used nowadays

In the topology above R1, R2 and R3 should run an IGP to communicate with each other because they are in the same AS. But to connect with other routers in another AS (like a different ISP), R1 and R3 must use an EGP.

With BGP, the term *autonomous system* (AS) refers to a network that operates separately from other networks and usually operates within a single administrative domain. Each AS is represented by an AS number. It is similar to EIGRP AS in this aspect. BGP is used mainly by the Internet Service Provider (ISP) all over the world. Each ISP usually has one BGP AS number (some very big ISP may have a few AS numbers). BGP AS numbers can be between 1 to 65,535.

In the topology above R1 and R3 are operating in BGP AS 1. If an AS connects to the public Internet using an EGP, then it must be assigned a unique AS number which is managed by the Internet Assigned Numbers Authority (IANA). IANA manages the AS numbers from **1 to 64,512** for public use (similar to public IP addresses) while **64,512 to 65,535** numbers are reserved for private use (similar to private IP addresses).

If we don't want to show the routers inside each AS we can simply ignore them:



In fact, the Internet that we are going “online” everyday is a collection of interconnected autonomous systems and BGP is running to provide routing between them.

Other BGP terms that you should learn are listed below:

+ **BGP speaker**: a router running BGP

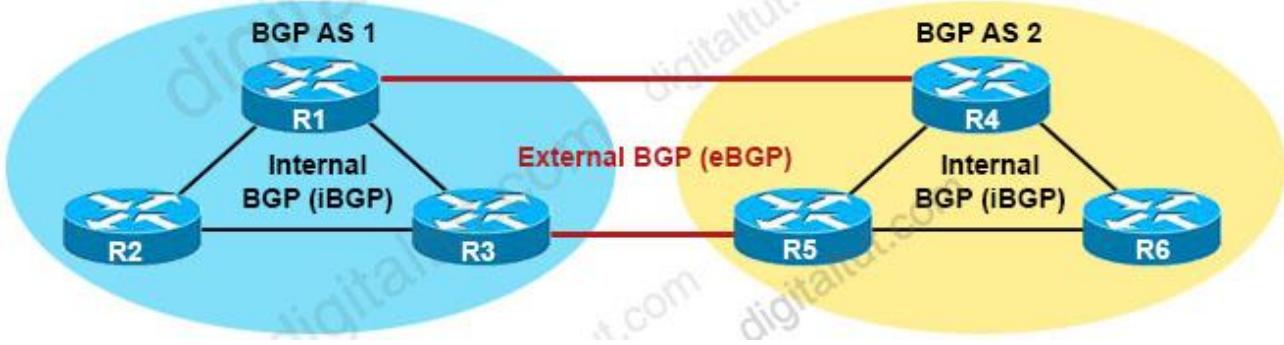
+ **BGP peer or BGP neighbor**: Any two routers that have formed a TCP connection to exchange BGP routing information (as BGP runs over TCP on port 179, not UDP)

+ **Prefix**: Maybe you learned the word “subnet”. In BGP world, it is usually called “prefix” because BGP usually does not advertise small subnets. It advertises blocks of large subnets so “prefix” is often used

instead

- + **Internal BGP (iBGP)**: refers to the BGP neighbor relationship within the same AS. The iBGP neighbor does not have to be directly connected
- + **External BGP (eBGP)**: refers to the BGP neighbor relationship between two peers belongs to different AS. It is recommended that eBGP should be directly connected. Never run an IGP between eBGP peers.

In the below topology suppose all routers are running BGP then R1 is considered internal BGP to R2 and R3 (as they are running same AS 1) but is external BGP to R4. R5 is internal to R4 and R6 but external to R3.

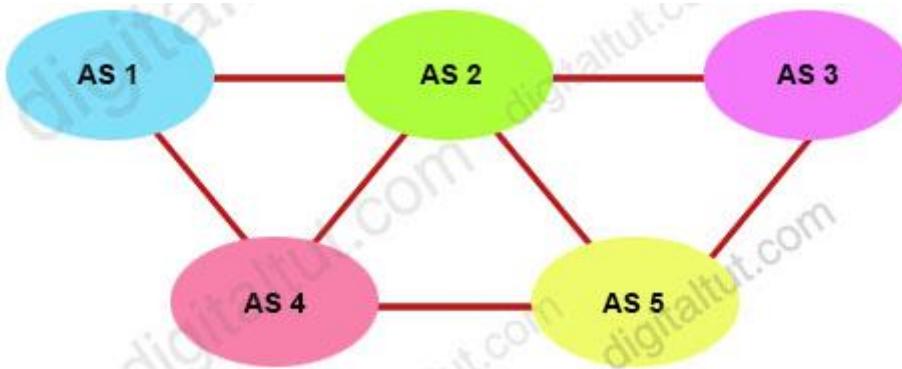


Why do we need BGP?

Maybe you will raise a question: "Why can't we only use OSPF or EIGRP instead?"

The most important reason is BGP greatly supports path control.

Maybe you learned and understood about EIGRP, OSPF routing protocols. They are different but both of them have the same purpose: find the most optimal path to the destination. But when we are working in ISP level we don't care much about this. In ISP we really want to control the path, even it is not the most optimal path, to the destination. For example, how can you choose which path to go from AS 1 to AS 3 in the topology below?



Suppose an IGP (like OSPF) is using all the way. With default values and same bandwidth on all the links, OSPF will choose the path AS 1 -> AS 2 -> AS 3. In order to manipulate the path we will have to change the bandwidth of some interfaces on the edge routers of each AS. A few months later we want to change the path then we have to configure the bandwidth again. With BGP we can easily define the path like this: "to go from AS 1 to AS 3 we will go through AS 2 then AS 4 then AS 5". Moreover we can easily control the ratio of traffic passing through each link. BGP can do this because it has a rich set of features to control the paths to the destination. One of the most popular features BGP uses to control the path is known as "attributes". However, discussion about "attributes" is out of the scope of this tutorial, which is dedicated for CCNA learners.

In fact, **BGP is a path vector protocol**. Path vector protocol does not rely on the bandwidth of the links (like OSPF) or hop count (like RIP) or a group of parameters (like EIGRP). Path vector protocol

relies on the number of autonomous systems it has to go through. In other words, it chooses the path with least number of autonomous systems (shortest AS Path) to reach the destination, provided that the path is loop-free. Of course we can change the path easily for our purpose.

For example in the topology above, if all the values are default and same, BGP will choose the path AS 1 -> AS 2 -> AS 3 for traffic from AS 1 destined to AS 3 because it only has to go through one AS (AS 2).

Another reason to use BGP is **BGP can handle very big routing tables**. The ISP level of routing does have large number of routes, which IGP cannot handle. BGP handles such large routes between AS. Currently the global Internet routing table contains over 500,000 routes.

Another good reason to use BGP is because **the great benefits of MPLS-based virtual private networks (MPLS VPN)**. MPLS VPN is beyond the scope of this tutorial but in general MPLS VPN provides traffic separation and path isolation on a shared network infrastructure and BGP is the background layer so that VPN can operate well. For example your corporation is running Voice over IP (VoIP), Video on Demand (VoD), Internet service... on the same infrastructure then you can use MPLS VPN to separate these services as if they were running on their own infrastructure.

In the next part we will learn about how your edge routers of your company can connect to the ISP.

Type of connection to ISP

BGP is often used to connect to the ISP so we list here all the type of connection to the ISP.

Single homed

Your company may connect to ISP in several ways. The most popular and simple way is single homed with a single link between the company and the ISP. With this design, only one possible next-hop router exists for all routes to the Internet.



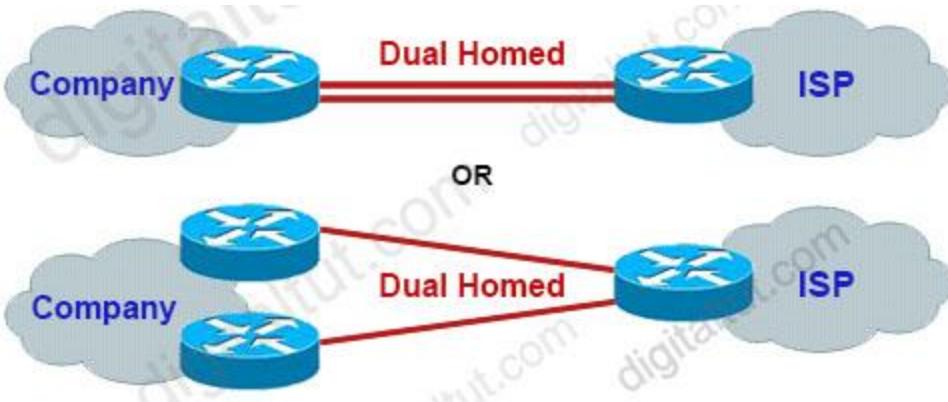
A big disadvantage of this design is when the link fails or either of the routers fails, the connection to the Internet fails as well. But of course, this design saves money comparing to multiple connections to the Internet designs and in fact it is the only reason for small company to accept this design.

With this design we don't need BGP in fact, all things we need are:

- + A default route from the company to the ISP
- + A static route from the ISP to the company's public address range

Dual homed

The next design is called "dual homed", in which the "dual" word refers to the designs with two links to the same router.

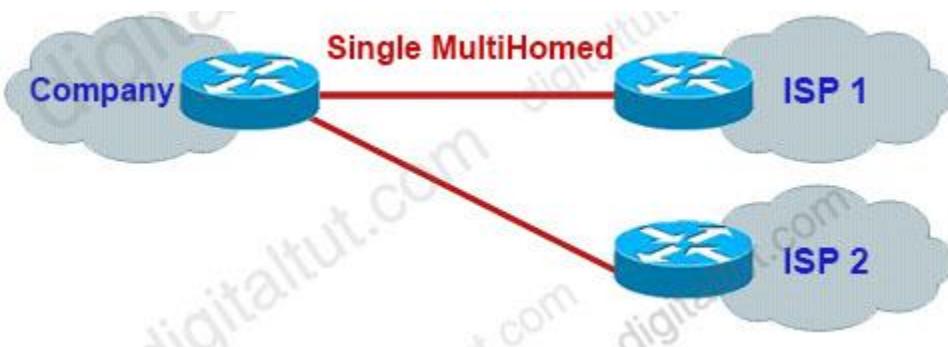


In this design we can use BGP to share the traffic between two routers of the company with our specific ratio (load balancing) or fail over. Of course this design is better in redundancy than the first one but it still has a "single point of failure" at the ISP router.

Single Multihomed

The next design is called "single multihomed" refers to:

- + Having connections to multiple ISPs from one router at the company
- + Single link per ISP.

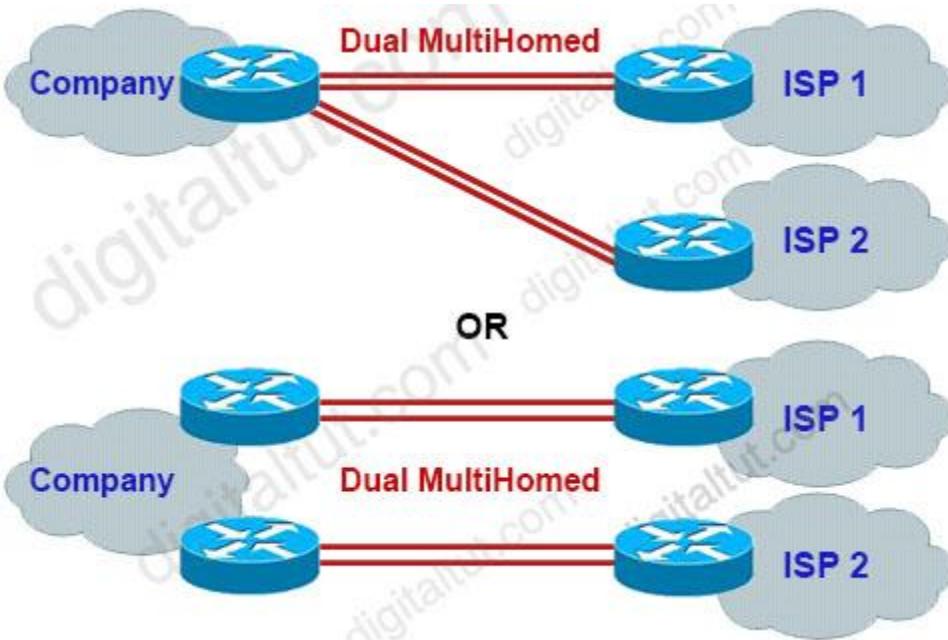


This design is good if we want to separate important traffic to a specific ISP while still has the other ISP as the fail over path.

Dual Multihomed

And the last design is called "dual multihomed" refers to:

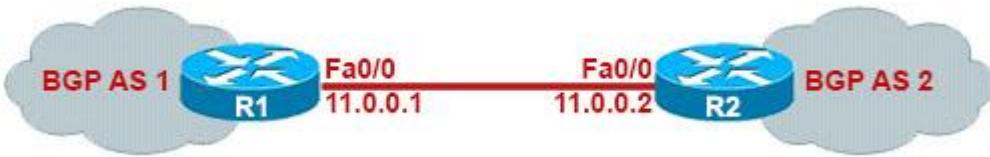
- + Multiple links per ISP
- + Multiple links to Company



If your company has a strong budget then Dual Multihomed design is ideal to make sure your connection to outside is always up. And BGP is highly recommended in this case.

In conclusion, except Single Homed design, BGP can be used effectively to control the traffic between your company/corporation to ISPs.

In this lab we will learn a simple eBGP (two BGP routers with different Autonomous System numbers) configuration between two routers with the topology below:



First we need to configure some interfaces on two routers as follows:

<pre>R1(config)#interface fastethernet0/0 R1(config-if)#ip address 11.0.0.1 255.255.255.0 R1(config-if)#no shutdown R1(config-if)#interface loopback 0 R1(config-if)#ip address 1.1.1.1 255.255.255.0</pre>	<pre>R2(config)#interface fastethernet0/0 R2(config-if)#ip address 11.0.0.2 255.255.255.0 R2(config-if)#no shutdown R2(config-if)#interface loopback 0 R2(config-if)#ip address 2.2.2.2 255.255.255.0</pre>
---	---

So we have just configured interface fa0/0 and loopback0 on both routers. Next we will configure the BGP configuration part on R1:

<pre>R1(config)#router bgp 1 R1(config-router)#neighbor 11.0.0.2 remote-as 2</pre>
--

The configuration is very simple with only two lines on R1. In the first line, BGP configuration begins with a familiar type of command: the **router bgp** command, where **AS number** is the BGP AS number used by that router (same as EIGRP, OSPF configuration).

The next command defines the IP address of the neighbor. Unlike OSPF or EIGRP, BGP cannot discover its neighbors automatically so we have to explicitly declare them. We also have to know and declare the neighbor's BGP AS number as well. In this case R1 wants to establish BGP neighbor relationship with R2 (in BGP AS 2) so it choose an interface on R2 (Fa0/0: 11.0.0.2) and specify R2 is in **BGP AS 2** via the

command “neighbor **11.0.0.2** remote-as **2**”. At the other end R2 will do the same thing for R1 to set up BGP neighbor relationship.

```
R2(config)#router bgp 2
R2(config-router)#neighbor 11.0.0.1 remote-as 1
```

After a moment we should see a message (on each router) similar to the following, letting us know that an adjacency has been formed:

On R1:

```
*Aug 17 00:09:38.453: %BGP-5-ADJCHANGE: neighbor 11.0.0.2 Up
```

On R2:

```
*Aug 17 00:09:38.453: %BGP-5-ADJCHANGE: neighbor 11.0.0.1 Up
```

So after forming BGP neighbor relationship we can verify by using the “show ip bgp summary” command on both routers:

```
R1#show ip bgp summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor      V     AS MsgRcvd MsgSent      TblVer  InQ OutQ Up/Down  State/PfxRcd
11.0.0.2      4      2    19      19          1       0     0 00:16:21      0

R2#show ip bgp summary
BGP router identifier 2.2.2.2, local AS number 2
BGP table version is 1, main routing table version 1

Neighbor      V     AS MsgRcvd MsgSent      TblVer  InQ OutQ Up/Down  State/PfxRcd
11.0.0.1      4      1    20      20          1       0     0 00:17:13      0
```

Please pay attention to the “State/PfxRcd” column of the output. It indicates the number of prefixes that have been received from a neighbor. If this value is a number (including “0”, which means BGP neighbor does not advertise any route) then the BGP neighbor relationship is good. If this value is a word (including “Idle”, “Connect”, “Active”, “OpenSent”, “OpenConfirm”) then the BGP neighbor relationship is not good.

In the outputs above we see the BGP neighbor relationship between R1 & R2 is good with zero Prefix Received (PfxRcd) because they have not advertised any routes yet.

How about the BGP routing table? We can check with the “show ip bgp” command but currently this table is empty! This is because although they formed BGP neighbor relationship but they have not exchanged any routes. Let’s try advertising the loopback 0 interface on R1 to R2:

```
R1(config-router)#network 1.1.1.0 mask 255.255.255.0
```

As you see, unlike other routing protocols like OSPF or EIGRP, we have to use subnet mask (255.255.255.0 in this case), not wildcard mask, to advertise the routes in the “network” command.

Note: With BGP, you must advertise the correct network and subnet mask in the “network” command (in this case network 1.1.1.0/24). BGP is very strict in the routing advertisements. In other words, BGP only advertises the network which exists exactly in the routing table (in this case network 1.1.1.0/24 exists in the routing table as the loopback 0 interface). If you put the command “network 1.1.0.0 mask 255.255.0.0” or “network 1.0.0.0 mask 255.0.0.0” or “network 1.1.1.1 mask 255.255.255.255” then BGP will not advertise anything.

Now the BGP routing tables on these two routers contain this route:

```
R1#sh ip bgp
BGP table version is 4, local router ID is 11.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf  Weight     Path
*> 1.1.1.0/24        0.0.0.0                  0          32768    i

R2#sh ip bgp
BGP table version is 2, local router ID is 11.0.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf  Weight     Path
*> 1.1.1.0/24        11.0.0.1                 0          0       1 i
```

An asterisk (*) in the first column means that the route has a valid next hop. A greater-than sign (>) indicates the route has been selected as the best path to that network.

The "Metric" column here is not the usual metric like in OSPF or EIGRP. It is the Multi Exit Discriminator (MED) attribute of BGP. "Weight" is another BGP attribute. The default values of both MED and Weight are 0 (as you see at the outputs above).

The "Path" column shows the AS paths that prefix were sent to reach us. It would better to read the "Path" from right to left to understand which path this prefix travel to reach our router. Letter "i" is considered the starting point of the prefix and the next number is the originating AS where this prefix originated. Next numbers are the recorded paths it traveled. For example if a prefix had to travel from AS 1 -> 2 -> 3 -> 4 -> 5 (our AS) then we will see the path "4 3 2 1 i" on our router.

Note: A blank AS path (only letter "i" is shown) means that the route was originated in the local AS. In the R1 output above, network 1.1.1.0/24 is originated from R1 so we see the path only has one letter "i".

One notice is on R1 the "Next Hop" is 0.0.0.0 which means this prefix is originated from the local router. On R2 the Next Hop is pointing toward the interface Fa0/0 of R1 (11.0.0.1) to which R2 will send traffic for the destination 1.1.1.0/24.

Now R1 advertised prefix 1.1.1.0/24 to R2 so we can re-check R2 with the "show ip bgp summary" command to see the "Prefix received" increased to 1:

```
R2#sh ip bgp summary
BGP router identifier 2.2.2.2, local AS number 2
BGP table version is 2, main routing table version 2
1 network entries using 117 bytes of memory
1 path entries using 52 bytes of memory
2/1 BGP path/bestpath attribute entries using 248 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 441 total bytes of memory
BGP activity 1/0 prefixes, 1/0 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
11.0.0.1	4	1	5	4	2	0	0	00:01:36	1

Also in the routing table of R2 we will see this prefix, which is advertised with BGP from R1:

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
```

Gateway of last resort is not set

```
1.0.0.0/24 is subnetted, 1 subnets
B      1.1.1.0 [20/0] via 11.0.0.1, 00:00:20
2.0.0.0/24 is subnetted, 1 subnets
C      2.2.2.0 is directly connected, Loopback0
11.0.0.0/24 is subnetted, 1 subnets
C      11.0.0.0 is directly connected, FastEthernet0/0
```

14. Layer 3 – RIP

Routing Information Protocol (RIP) is a distance-vector routing protocol which is based on Bellman-Ford algorithm. Routers using Distance Vector routing protocols do not possess the topological information about the network but instead rely on the neighbors information (so this method is known as routing by rumor). RIP sends the complete routing table out to all active interfaces every 30 seconds. RIP only uses hop count (the number of routers) to determine the best way to a remote network.

Note: RIP v1 is a classful routing protocol but RIP v2 is a classless routing protocol.

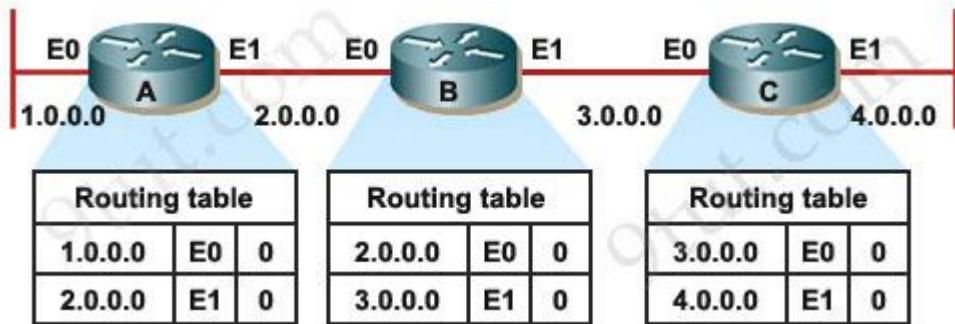
Classful routing protocols do not include the subnet mask with the network address in routing updates, which can cause problems with discontiguous subnets or networks that use Variable-Length Subnet Masking (VLSM). Fortunately, RIPv2 is a classless routing protocol so subnet masks are included in the routing updates, making RIPv2 more compatible with modern routing environments.

Distance vector protocols advertise routing information by sending messages, called routing updates, out the interfaces on a router.

RIP Operation

A big problem with distance vector routing protocol is routing loop. Let's take a look at how a routing loop occurs.

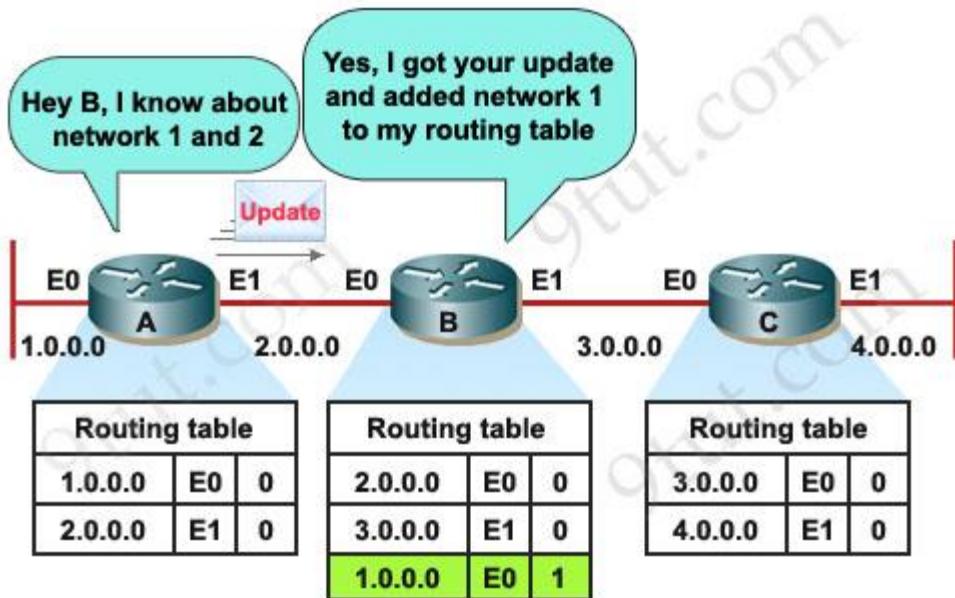
Here we have routers A, B and C. Notice that at the beginning (when a routing protocol is not turned on) there are only directly connected networks in the routing tables of these routers. For example, in the routing table of router A, **network 1.0.0.0** has already been known because it is directly connected through **interface E0** and the **metric** (of a directly connected network) is **0** (these 3 parameters are shown in the routing tables below).



Also B knows networks **2.0.0.0** & **3.0.0.0** with a **metric of 0**.
Also C knows networks **3.0.0.0** & **4.0.0.0** with a **metric of 0**.

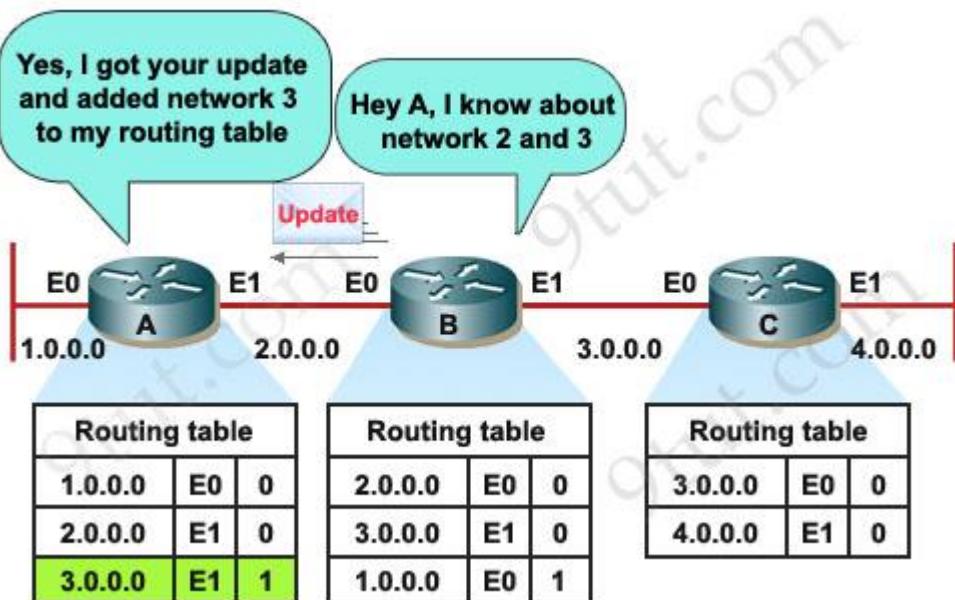
Now we turn on RIP on these routers (we will discuss the configuration later. In the rest of this article, we will call network 1.0.0.0 network 1, 2.0.0.0 network 2 and so on).

RIP sends update every 30 seconds so after 30 sec goes by, A sends a copy of its routing table to B, B already knew about network 2 but now B learns about network 1 as well. Notice the metric we have here for directly connected networks, since we're using RIP, we're using a metric of hop count. Remember a hop count (or a hop) is how many routers that these packets will have to go through to reach the destination. For example, from router A to network 1 & 2 (which are directly connected) it goes to 0 hop, router B has now learned about network 1 from A via E0 interface so the metric now will be 1 hop.

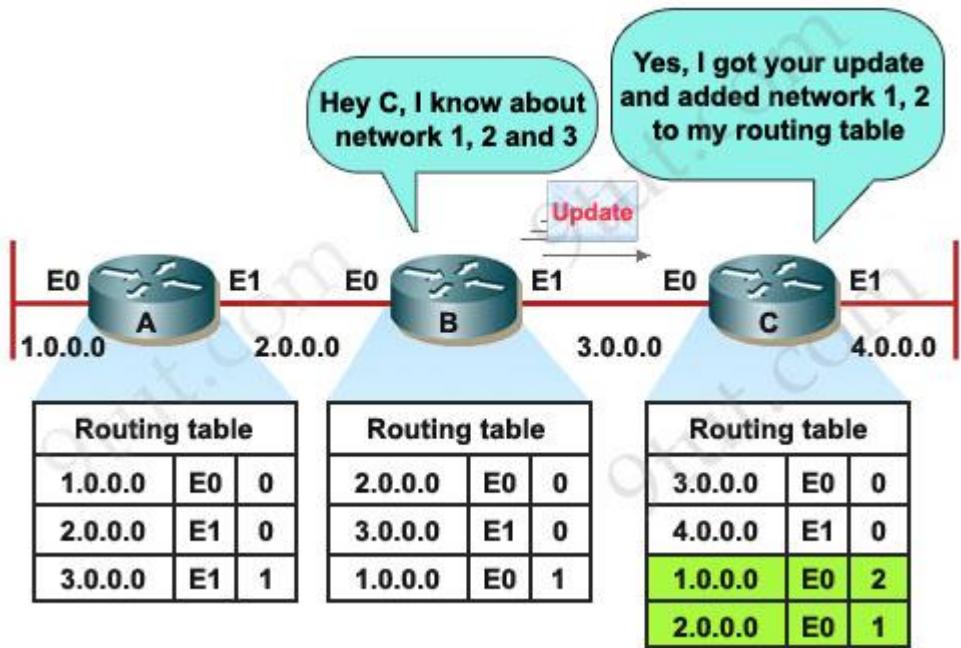


Each router receives a routing table from its direct neighbor. For example, Router B receives information from Router A about network 1 and 2. It then adds a distance vector metric (such as the number of hops), increasing the distance vector of these routes by 1.

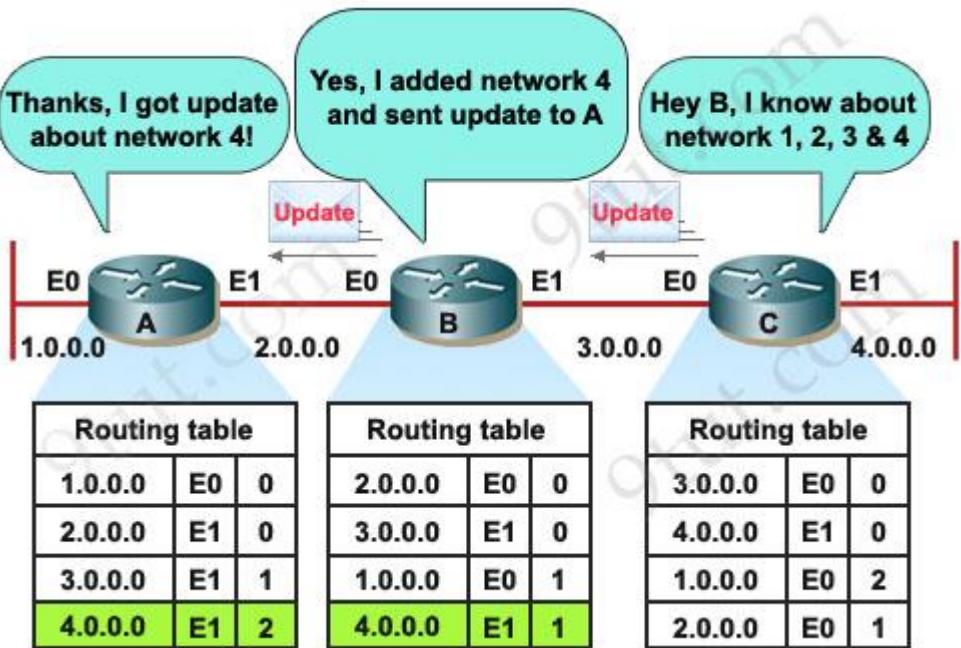
B also exchanges its routing table with A about network 2 and 3.



B then passes the routing table to its other neighbor, Router C.



C also sends its update to B and B sends it to A.



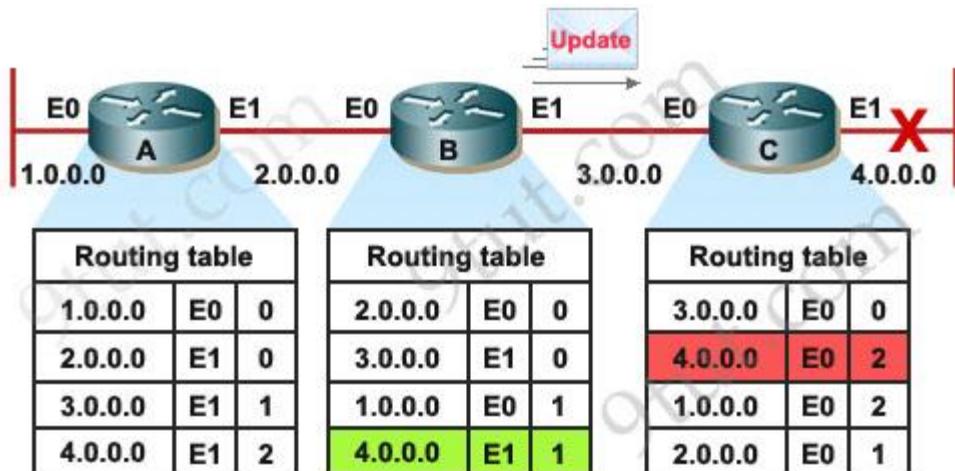
Now the network is converged.

Now let's assume network 4 down suddenly.

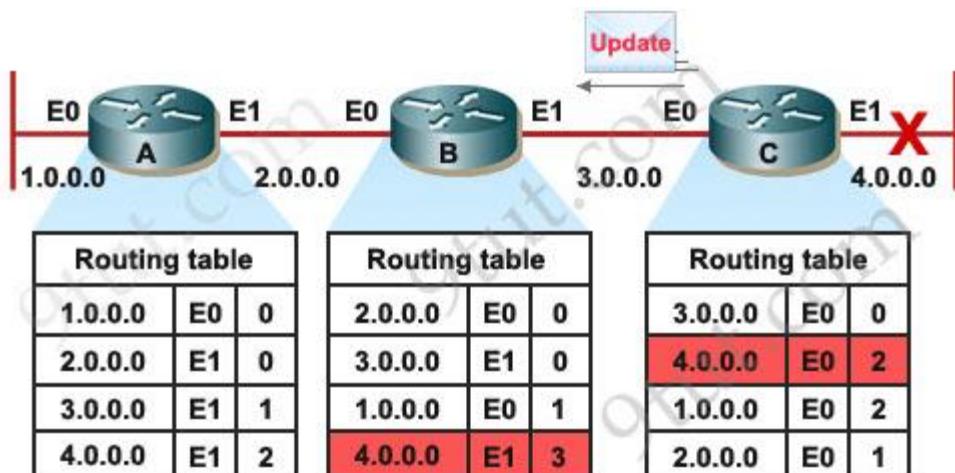


When network 4 fails, Router C detects the failure and stops routing packets out its E1 interface. However, Routers A and B have not yet received notification of the failure. Router A still believes it can access 4.0.0.0 through Router B. The routing table of Router A still reflects a path to network 10.4.0.0 with a distance of 2 and router B has a path with a distance of 1.

There will be no problem if C sends an update earlier than B and inform that network is currently down but if B sends its update first, C will see B has a path to network 4 with a metric of 1 so it updates its routing table, thinking that "if B can go to network 4 by 1 hop than I can go to network 4 by 2 hops" but of course this is totally wrong.



The problem does not stop here. In turn, C sends an update to B and informs it can access network 4 by 2 hops. B learns this and think "if C can access network 4 by 2 hops than I can access by 3 hops".



This same process occurs when B continually sends its update to C and the metric will increase to infinity so this phenomenon is called "counting to infinity".

Below lists some methods to prevent this phenomenon:

SPLIT HORIZON:

A router never sends information about a route back in same direction which is original information came, routers keep track of where the information about a route came from. Means when router A sends update to router B about any failure network, router B does not send any update for same network to router A in same direction.

ROUTE POISONING:

Router consider route advertised with an infinitive metric to have failed (metric=16) instead of marking it down. For example, when network 4 goes down, router C starts route poisoning by advertising the metric (hop count) of this network as 16, which indicates an unreachable network. When router B receives this advertising, it continue advertising this network with a metric of 16.

POISON REVERSE:

The poison reverse rule overwrites split horizon rule. For example, if router B receives a route poisoning of network 4 from router C then router B will send an update back to router C (which breaks the split horizon rule) with the same poisoned hop count of 16. This ensures all the routers in the domain receive the poisoned route update.

Notice that every router performs poison reverse when learning about a downed network. In the above example, router A also performs poison reverse when learning about the downed network from B.

HOLD DOWN TIMERS:

After hearing a route poisoning, router starts a hold-down timer for that route. If it gets an update with a better metric than the originally recorded metric within the hold-down timer period, the hold-down timer is removed and data can be sent to that network. Also within the hold-down timer, if an update is received from a different router than the one who performed route poisoning with an equal or poorer metric, that update is ignored. During the hold-down timer, the "downed" route appears as "possibly down" in the routing table.

For example, in the above example, when B receives a route poisoning update from C, it marks network 4 as "possibly down" in its routing table and starts the hold-down timer for network 4. In this period if it receives an update from C informing that the network 4 is recovered then B will accept that information, remove the hold-down timer and allow data to go to that network. But if B receives an update from A informing that it can reach network by 1 (or more) hop, that update will be ignored and the hold-down timer keeps counting.

Note: The default hold-down timer value = 180 second.

TRIGGERED UPDATE :

When any route failed in network ,do not wait for the next periodic update instead send an immediate update listing the poison route.

COUNTING TO INFINITY:

Maximum count 15 hops after it will not be reachable.

RIP Timers

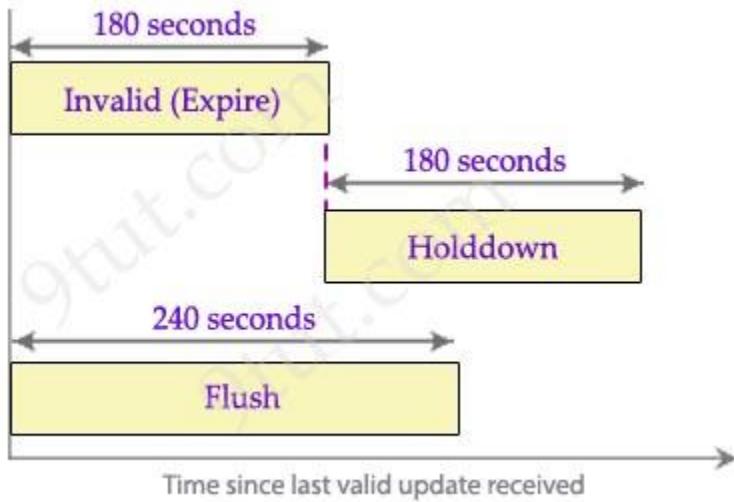
RIP uses several timers to regulate its operation. These timers are described below:

Update timer: how often the router sends update. Default update timer is 30 seconds

Invalid timer (also called **Expire timer**): how much time must expire before a route becomes invalid since seeing a valid update; and place the route into holddown. Default invalid timer is 180 seconds

Holddown timer: When a route is expired, it enters "holddown", which means the router will not believe any new updates with a hop count equal to or higher (poorer) than the hop count recording in the routing table. Hold down is intended to assist in avoiding inaccurate routing by rumor information while the network converges. Default holddown timer is 180 seconds

Flush timer: how much time since the last valid update, until RIP deletes that route in its routing table. Default Flush timer is 240 seconds



Note: From the image above, you can see that when a route expires, in fact the Holddown timer only works in its first 60 seconds (not 180 seconds), then the route is removed from the routing table when the Flush timer is expired.

15. Layer 3 – OSPF

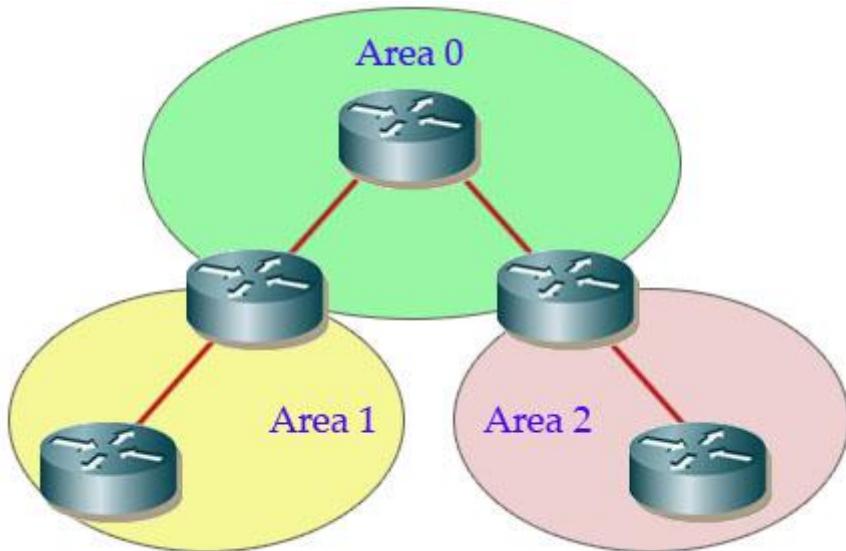
In this article we will learn about the OSPF Routing Protocol

Open-Shortest-Path-First (OSPF) is the most widely used interior gateway protocol routing protocol on the world because it is a public (non-proprietary) routing protocol while its biggest rival, EIGRP, is a Cisco proprietary protocol so other vendors can't use it (edit: EIGRP has become a public routing protocol since 2013). OSPF is a complex link-state routing protocol. Link-state routing protocols generate routing updates only when a change occurs in the network topology. When a link changes state, the device that detected the change creates a link-state advertisement (LSA) concerning that link and sends to all neighboring devices using a special multicast address. Each routing device takes a copy of the LSA, updates its link-state database (LSDB), and forwards the LSA to all neighboring devices.

Note:

- + OSPF routers use LSA (Link State Advertisement) to describe its link state. LSDB stores all LSAs.
- + A router uses Router LSA to describe its interface IP addresses.
- + After OSPF is started on a router, it creates LSDB that contains one entry: this router's Router LSA.

There are five types of OSPF Link-State Packets (LSPs).



- + **Hello:** are used to establish and maintain adjacency with other OSPF routers. They are also used to elect the Designated Router (DR) and Backup Designated Router (BDR) on multiaccess networks (like Ethernet or Frame Relay).
- + **Database Description** (DBD or DD): contains an abbreviated list of the sending router's link-state database and is used by receiving routers to check against the local link-state database
- + **Link-State Request** (LSR): used by receiving routers to request more information about any entry in the DBD
- + **Link-State Update** (LSU): used to reply to LSRs as well as to announce new information. LSUs contain seven different types of Link-State Advertisements (LSAs)
- + **Link-State Acknowledgement** (LSAck): sent to confirm receipt of an LSU message

Key points

- + Is a public (non-proprietary) routing protocol.

- + Is the only link-state routing protocol you learn in CCNA
- + This works by using the Dijkstra algorithm
- + Information about its neighbors (local connectivity) is sent to the entire network using multicasting
- + Routing information is shared through Link-state updates (LSAs)
- + HELLO messages are used to maintain adjacent neighbors. By default, OSPF routers send Hello packets every 10 seconds on multiaccess and point-to-point segments and every 30 seconds on non-broadcast multiaccess (NBMA) segments (like Frame Relay, X.25, ATM).
- + Is a classless routing protocol because it does not assume the default subnet masks are used. It sends the subnet mask in the routing update.
- + Supports VLSM and route summarization
- + Uses COST as a metric which CISCO defines as the inverse of the bandwidth
- + Uses AREAs to subdivide large networks, providing a hierarchical structure and limit the multicast LSAs within routers of the same area — Area 0 is called **backbone area** and all other areas connect directly to it. All OSPF networks must have a backbone area
- + Only support IP but it's not bad as we are all using IP, right? :)

Area Border Routers (ABR) are any routers that have one interface in one area and another interface in another area

Let's see an example of OSPF

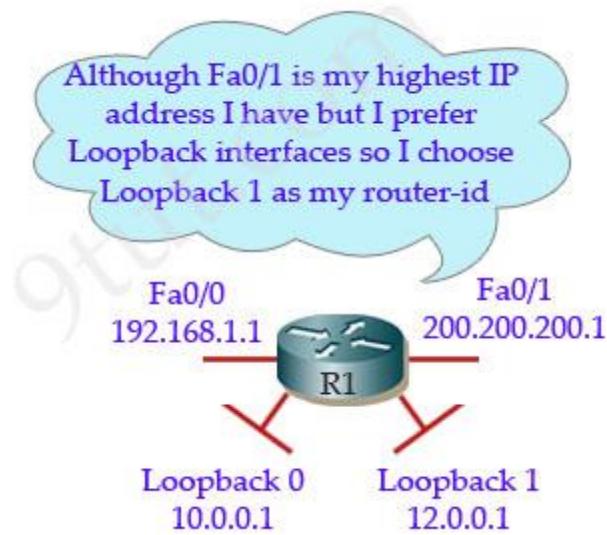
Suppose OSPF has just been enabled on R1 & R2. Both R1 and R2 are very eager to discover if they have any neighbors nearby but before sending Hello messages they must first choose an OSPF router identifier (router-id) to tell their neighbors who they are. The Router ID (RID) is an IP address used to identify the router and is chosen using the following sequence:

- + The highest IP address assigned to a loopback (logical) interface.
- + If a loopback interface is not defined, the highest IP address of all active router's physical interfaces will be chosen.
- + The router ID can be manually assigned

In this example, suppose R1 has 2 loopback interfaces & 2 physical interfaces:

- + Loopback 0: 10.0.0.1
- + Loopback 1: 12.0.0.1
- + Fa0/0: 192.168.1.1
- + Fa0/1: 200.200.200.1

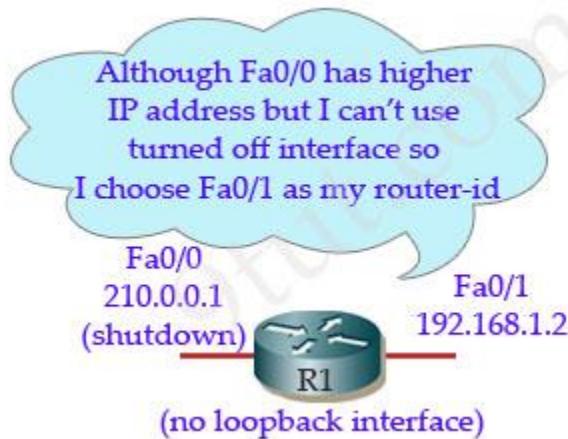
As said above, the loopback interfaces are preferred to physical interfaces (because they are never down) so the highest IP address of the loopback interfaces is chosen as the router-id -> Loopback 1 IP address is chosen as the router-id.



Suppose R1 doesn't have any loopback interfaces but it has 2 physical interfaces:

- + Fa0/0: 210.0.0.1 but it is shutdown
- + Fa0/1: 192.168.1.2 (is active)

Although Fa0/0 has higher IP address but it is shutdown so R1 will choose Fa0/1 as its router-id.



Now both the routers have the router-id so they will send Hello packets on all OSPF-enabled interfaces to determine if there are any neighbors on those links. The information in the OSPF Hello includes the OSPF Router ID of the router sending the Hello packet.

For example, R1 wants to find out if it has any neighbor running OSPF it sends a Hello message to the multicast address 224.0.0.5. This is the multicast address for all OSPF routers and all routers running OSPF will proceed this message.



If an OSPF router receives an OSPF Hello packet that satisfied all its requirement then it will establish adjacency with the router that sent the Hello packet. In this example, if R1 meet R2's requirements, meaning it has the same **Hello interval**, **Dead interval** and **AREA number**, R2 will add R1 to its neighbor table.

- + **Hello interval:** indicates how often it sends Hello packets. By default, OSPF routers send Hello packets every 10 seconds on multiaccess and point-to-point segments and every 30 seconds on non-broadcast multiaccess (NBMA) segments (like Frame Relay, X.25, ATM)
- + **Dead interval:** number of seconds this router should wait between receiving hello packets from a neighbor before declaring the adjacency to that neighbor down
- + **AREA number:** the area it belongs to



Now R1 and R2 are neighbors but they don't exchange LSAs immediately. Instead, they sends Database Description (DD or DBD) packets which contain an abbreviated list of the sending router's link-state database.

The neighbors also determine who will be the master and who will be the slave. The router which higher router-id will become master and initiates the database exchange. The receiver acknowledges a received DD packet by sending an identical DD packet back to the sender. Each DD packet has a sequence number and only the master can increment sequence numbers.

We are now neighbors, cheers!
Now let's exchange our databases
I have higher Router-ID so I will
be master and send DD first



I have lower Router-ID
so I will be slave. Here is my DD



R1 or R2 can send Request to get missing LSA from its neighbors

Hey, I need to know more
about some of your routes!





R2 sends back an LSAck packet to acknowledge the packet



There are 3 type of tables

- + Neighbor
- + Topology
- + Routing

Neighbor table

- + Contain information about the neighbors
- + Neighbor is a router which shares a link on same network
- + Another relationship is adjacency
- + Not necessarily all neighbors
- + LSA updates are only when adjacency is established

Topology table

- + Contain information about all network and path to reach any network
- + All LSA's are entered into the topology table
- + When topology changes LSA's are generated and send new LSA's
- + On topology table an algorithm is run to create a shortest path, this algorithm is known as SPF or dijkstra algorithm

Routing Table

- + Also known as forwarding database
- + Generated when an algorithm is run on the topology database
- + Routing table for each router is unique

D: Exchange LSDB's list

Neighbors use DD (Data Description) to exchange their LSDB catalogs. In this scenario, R1 sends DD to R2 first. It says: I have a Route LSA from R1. R2 also sends DD to R1: I have a Route LSA from R2.

Note: DD works like table of contents. It lists what LSDB has, but not details. By reading DD, the receiving router can determine what it is missing and then ask the sender to transmit required LSAs..

R1 Request, R2 Update

R1 has learned that R2 has a R2 Router LSA that it does not have.

R1 sends a LS Request to R2. When R2 receives this request, it sends an Update to transmit this LSA to R1.

R2 Request, R1 Update

R2 also sends request to R1. R1 replies an Update. Upon receiving Update, R2 adds R1 Router LSA to its LSDB, calculates its routes, and add a new entry (192.168.1.0, S1/0) to its routing table.

Note: OSPF works distributedly. After routers have synchronized their LSDB, they use the same data (LSDB) to calculate shortest paths, and updates their routing tables independently.

Ack update : LSAs are received

In order to assure reliable transmission, when a router receives an Update, it sends an Ack to the Update sender. If the sender does not receive Ack within a specific period, it times out and retransmits Update.

Note: OSPF uses Update-Ack to implement reliable transmission. It does not use TCP.

H1 ping H2: succeeded.

Each OSPF router creates a Router LSA to describe its interfaces' IP addresses and floods its Router LSA to its neighbors. After a few rounds of flooding, all OSPF routers have the same set of Router LSAs in their LSDBs. Now routers can use the same LSDB to calculate routes and update routing tables.

From LSDB, a router learns the entire topology: the number of routers being connected. Router interfaces and their IP addresses, interface link costs (OSPF metric). With such detailed information, routers are able to calculate routing paths to reach all destinations found in LSDB. For example, in the OSPF basic simulation (see External links), R1's LSDB contains two Router LSAs: – A Router LSA from R1. R1 has two links. Their IP addresses are 192.168.1.0/24, 192.168.3.0/30. – A Router LSA from R2. R2 has two links. Their IP addresses are 192.168.2.0/24, 192.168.3.0/30. From these LSAs, R1 can calculate the routing path to reach remote destination 192.168.2.2 and adds an entry (192.168.2.0/24, S1/0) to its routing table.

16. Layer 3 – IPsec VPN

16.1 IPsec components

There are three main IPsec protocols that determine how IPsec modifies IP packets:

- *Internet Key Exchange (IKE)* establishes the SA between the communicating hosts, negotiating the cryptographic keys and algorithms that will be used in the course of the session.
- *Authentication Header (AH)* adds a header field to the packet being sent that includes a cryptographic hash of the packet's contents. The host that receives the packet can use this hash to ensure that the payload hasn't been modified in transit.
- *Encapsulating Security Payload (ESP)* encrypts the payload. It also adds a sequence number to the packet header so that the receiving host can be sure it isn't getting duplicate packets.

Note that it's possible to use both AH and ESP simultaneously, although newer versions of the ESP protocol incorporate much of AH's functionality. At any rate, both protocols are built into IP implementations.

16.2 IPsec encryption

The encryption established by IKE and ESP does much of the work we expect out of an IPsec VPN. You'll notice that we've been a little vague about how the encryption works here; that's because IKE and IPsec permit a wide range of encryption suites and technologies to be used, which is why IPsec has managed to survive over more than two decades of advances in this area. It's fairly common for IPsec VPNs to use Public Key Infrastructure (PKI) for their cryptographic purposes, but that's by no means required and there are other options available.

16.3 IPsec modes: IPsec Tunnel vs. IPsec Transport

There are two different ways in which IPsec can operate, referred to as *modes*: Tunnel Mode and Transport Mode. The difference between the two pertains to how IPsec treats packet headers. In Transport Mode, IPsec encrypts (or authenticates, if only AH is being used) only the payload of the packet, but leaves the existing packet header data more or less as is. In Tunnel Mode, IPsec creates an entirely new packet with a new header, encrypts (or authenticates) the entire original packet including its header, and uses the modified original packet as the payload for the new packet.

When would you use the different modes? If a network packet has been sent from or is destined for a host on a private network, that packet's header includes routing data about those networks—and hackers can analyze that information and use it for nefarious purposes. Tunnel Mode, which protects that information, is generally used for connections between the gateways that sit at the outer edges of private corporate networks. A packet is encrypted as it leaves one network, and put inside a new packet whose destination is the gateway for the target network. Once it arrives at the gateway, it's decrypted and removed from the encapsulating packet, and sent along its way to the target host on the internal network. The header data about the topography of the private networks is thus never exposed while the packet traverses the public internet.

Transport mode, on the other hand, is generally used for workstation-to-gateway and direct host-to-host connections. A service tech using Windows Remote Desktop to help diagnose a problem on a user's computer would be using a Transport Mode connection, for instance.

16.4 IPsec VPNs vs. SSL VPNs

As noted above, IPsec VPNs are not the only game in town. There are also SSL VPNs, which as the name implies are secured by the TLS protocol rather than IPsec. SSL VPNs operate via a web browser and are generally used for accessing firewalled intranet websites. Because they're built into the browser software that everyone's familiar with, SSL VPNs are much simpler to use; IPsec VPNs often require the installation and configuration of specialized software. SSL VPNs can also provide more granularly limited access to the private network.

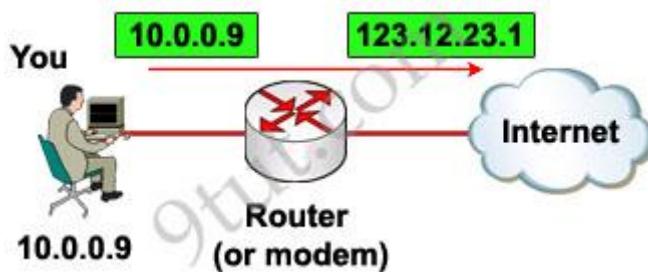
On the other hand, because it uses TLS, an SSL VPN is secured at the transport layer, not the network layer, so that may affect your view of how much it enhances the security of your connection.

17. Layer 3 – NAT

To go to the Internet we need to get an public IP address and it is unique all over the world. If each host in the world required a unique public IP address, we would have run out of IP address years ago. But by using Network Address Translation (NAT) we can save tons of IP addresses for later uses. We can understand NAT like this:

"NAT allows a host that does not have a valid registered IP address to communicate with other hosts through the Internet"

For example your computer is assigned a private IP address of 10.0.0.9 and of course this address can not be routed on the internet but you can still access the internet. This is because your router (or modem) translates this address into a public IP address, 123.12.23.1 for example, before routing your data into the internet.



Of course when your router receives a reply packet destined for 123.12.23.1 it will convert back to your private IP 10.0.0.9 before sending that packet to you.

Maybe you will ask "hey, I don't see any difference of using NAT to save tons of IP addresses because you still need a public IP address for each host to access the Internet and it doesn't save you anything, why you need to use NAT?"

Ok, you are right :), in the above example we don't see its usefulness but you now understand the fundamental of NAT!

Let's take another example!

Suppose your company has 500 employees but your Internet Service Provider (ISP) only gives you 50 public IP addresses. It means that you can only allow 50 hosts to access the internet at the same time. Here NAT comes to save your life!

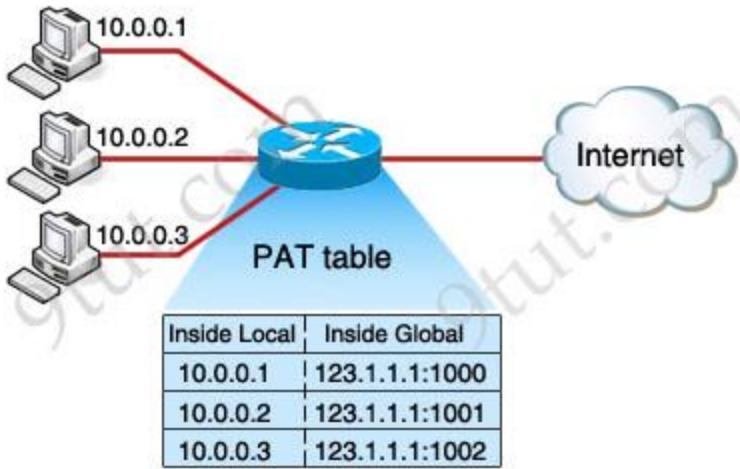
One thing you should notice that in real life, not all of your employees uses internet at the same time. Say, maybe 50 of them use internet to read newspaper at the morning; 50 others use internet at noon for checking mail... By using NAT you can dynamically assign these 50 public IP addresses to those who really need them at that time. This is called **dynamic NAT**.

But the above NAT solution does not solve our problem completely because in some days there can be more than 50 people surfing web at the morning. In this case, only the first 50 people can access internet, others must wait to their turns.

Another problem is, in fact, your ISP only gives you much lesser IP addresses than the number 50 because each public IP is very precious now.

To solve the two problems above, another feature of NAT can be used: **NAT Overload** or sometimes called **Port Address Translation (PAT)**

PAT permits multiple devices on a local area network (LAN) to be mapped to a single public IP address with different port numbers. Therefore, it's also known as port address translation (PAT). When using PAT, the router maintains unique source port numbers on the **inside global** IP address to distinguish between translations. In the below example, each host is assigned to the same public IP address 123.1.1.1 but with different port numbers (from 1000 to 1002).



Note: Cisco uses the term **inside local** for the private IP addresses and **inside global** for the public IP addresses replaced by the router.

The outside host IP address can also be changed with NAT. The **outside global** address represents the outside host with a public IP address that can be used for routing in the public Internet.

The last term, **outside local** address, is a private address of an external device as it is referred to by devices on its local network. You can understand outside local address as the inside local address of the external device which lies at the other end of the Internet.

Maybe you will ask how many ports can we use for each IP? Well, because the port number field has 16 bits, PAT can support about 2^{16} ports, which is more than 64,000 connections using one public IP address.

Now you have learned all the most useful features of NAT but we should summarize all features of NAT:

There are two types of NAT translation: dynamic and static.

Static NAT: Designed to allow one-to-one mapping between local and global addresses. This flavor requires you to have one real Internet IP address for every host on your network.

Dynamic NAT: Designed to map an unregistered IP address to a registered IP address from a pool of registered IP addresses. You don't have to statically configure your router to map an inside to an outside address as in static NAT, but you do have to have enough real IP addresses for everyone who wants to send packets through the Internet. With dynamic NAT, you can configure the NAT router with more IP addresses in the inside local address list than in the inside global address pool. When being defined in the inside global address pool, the router allocates registered public IP addresses from the pool until all are allocated. If all the public IP addresses are already allocated, the router discards the packet that requires a public IP address.

PAT (NAT Overloading): is also a kind of dynamic NAT that maps multiple private IP addresses to a single public IP address (many-to-one) by using different ports. Static NAT and Dynamic NAT both require a one-to-one mapping from the inside local to the inside global address. By using PAT, you can have thousands of users connect to the Internet using only one real global IP address. PAT is the technology that helps us not run out of public IP address on the Internet. This is the most popular type of NAT.

Besides NAT gives you the option to advertise only a single address for your entire network to the outside world. Doing this effectively hides the internal network from the public world really well, giving you some additional security for your network.

NAT terms:

- * **Inside local address** – The IP address assigned to a host on the inside network. The address is usually not an IP address assigned by the Internet Network Information Center (InterNIC) or service provider. This address is likely to be an RFC 1918 private address.
- * **Inside global address** – A legitimate IP address assigned by the InterNIC or service provider that represents one or more inside local IP addresses to the outside world.
- * **Outside local address** – The IP address of an outside host as it is known to the hosts on the inside

network.

* **Outside global address** – The IP address assigned to a host on the outside network. The owner of the host assigns this address.



To learn how to configure NAT please read my [Configure NAT GNS3 Lab tutorial](#)

18. Layer 4 – TCP

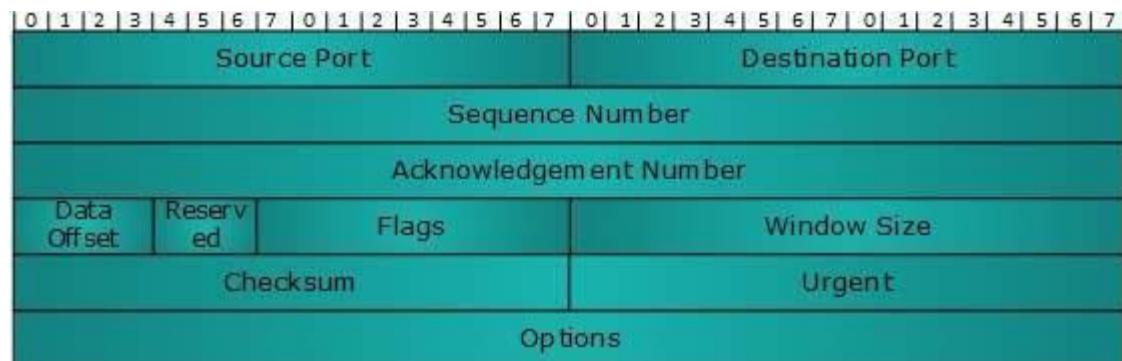
The transmission Control Protocol (TCP) is one of the most important protocols of Internet Protocols suite. It is most widely used protocol for data transmission in communication network such as internet.

18.1 Features

- TCP is reliable protocol. That is, the receiver always sends either positive or negative acknowledgement about the data packet to the sender, so that the sender always has bright clue about whether the data packet is reached the destination or it needs to resend it.
- TCP ensures that the data reaches intended destination in the same order it was sent.
- TCP is connection oriented. TCP requires that connection between two remote points be established before sending actual data.
- TCP provides error-checking and recovery mechanism.
- TCP provides end-to-end communication.
- TCP provides flow control and quality of service.
- TCP operates in Client/Server point-to-point mode.
- TCP provides full duplex server, i.e. it can perform roles of both receiver and sender.

18.2 Header

The length of TCP header is minimum 20 bytes long and maximum 60 bytes.



- **Source Port (16-bits)** - It identifies source port of the application process on the sending device.
- **Destination Port (16-bits)** - It identifies destination port of the application process on the receiving device.

- **Sequence Number (32-bits)** - Sequence number of data bytes of a segment in a session.
- **Acknowledgement Number (32-bits)** - When ACK flag is set, this number contains the next sequence number of the data byte expected and works as acknowledgement of the previous data received.
- **Data Offset (4-bits)** - This field implies both, the size of TCP header (32-bit words) and the offset of data in current packet in the whole TCP segment.
- **Reserved (3-bits)** - Reserved for future use and all are set zero by default.
- **Flags (1-bit each)**
 - NS - Nonce Sum bit is used by Explicit Congestion Notification signaling process.
 - CWR - When a host receives packet with ECE bit set, it sets Congestion Windows Reduced to acknowledge that ECE received.
 - ECE - It has two meanings:
 - If SYN bit is clear to 0, then ECE means that the IP packet has its CE (congestion experience) bit set.
 - If SYN bit is set to 1, ECE means that the device is ECT capable.
 - URG - It indicates that Urgent Pointer field has significant data and should be processed.
 - ACK - It indicates that Acknowledgement field has significance. If ACK is cleared to 0, it indicates that packet does not contain any acknowledgement.
 - PSH - When set, it is a request to the receiving station to PUSH data (as soon as it comes) to the receiving application without buffering it.
 - RST - Reset flag has the following features:
 - It is used to refuse an incoming connection.
 - It is used to reject a segment.
 - It is used to restart a connection.
 - SYN - This flag is used to set up a connection between hosts.
 - FIN - This flag is used to release a connection and no more data is exchanged thereafter. Because packets with SYN and FIN flags have sequence numbers, they are processed in correct order.

- **Windows Size** - This field is used for flow control between two stations and indicates the amount of buffer (in bytes) the receiver has allocated for a segment, i.e. how much data is the receiver expecting.
- **Checksum** - This field contains the checksum of Header, Data and Pseudo Headers.
- **Urgent Pointer** - It points to the urgent data byte if URG flag is set to 1.
- **Options** - It facilitates additional options which are not covered by the regular header. Option field is always described in 32-bit words. If this field contains data less than 32-bit, padding is used to cover the remaining bits to reach 32-bit boundary.

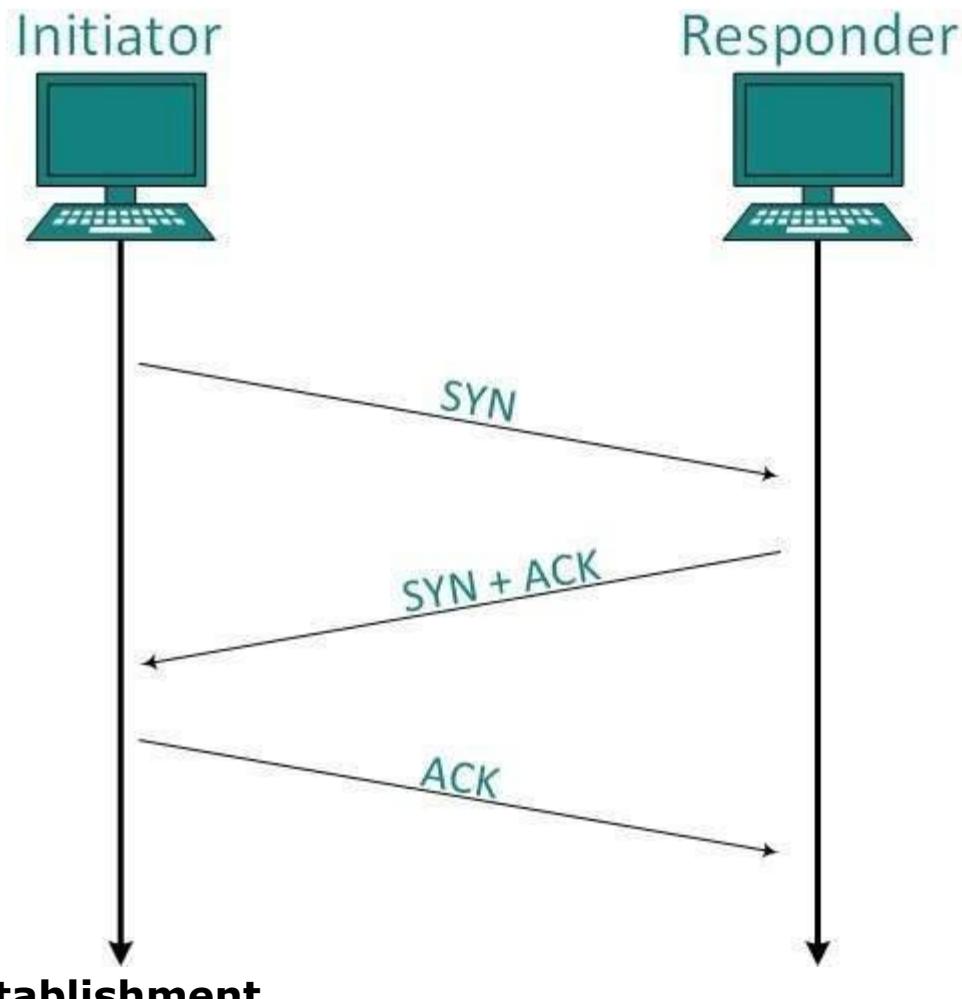
18.3 Addressing

TCP communication between two remote hosts is done by means of port numbers (TSAPs). Ports numbers can range from 0 – 65535 which are divided as:

- System Ports (0 – 1023)
- User Ports (1024 – 49151)
- Private/Dynamic Ports (49152 – 65535)

18.4 Connection Management

TCP communication works in Server/Client model. The client initiates the connection and the server either accepts or rejects it. Three-way handshaking is used for connection management.



Establishment

Client initiates the connection and sends the segment with a Sequence number. Server acknowledges it back with its own Sequence number and ACK of client's segment which is one more than client's Sequence number. Client after receiving ACK of its segment sends an acknowledgement of Server's response.

Release

Either of server and client can send TCP segment with FIN flag set to 1. When the receiving end responds it back by ACKnowledging FIN, that direction of TCP communication is closed and connection is released.

18.5 Bandwidth Management

TCP uses the concept of window size to accommodate the need of Bandwidth management. Window size tells the sender at the remote end, the number of data byte segments the receiver at this end can receive. TCP uses slow start phase by using window size 1 and increases the window size exponentially after each successful communication.

For example, the client uses windows size 2 and sends 2 bytes of data. When the acknowledgement of this segment received the windows size is doubled to 4 and next sent the segment sent will be 4 data bytes long. When the acknowledgement of 4-byte data segment is received, the client sets windows size to 8 and so on.

If an acknowledgement is missed, i.e. data lost in transit network or it received NACK, then the window size is reduced to half and slow start phase starts again.

18.6 Error Control &and Flow Control

TCP uses port numbers to know what application process it needs to handover the data segment. Along with that, it uses sequence numbers to synchronize itself with the remote host. All data segments are sent and received with sequence numbers. The Sender knows which last data segment was received by the Receiver when it gets ACK. The Receiver knows about the last segment sent by the Sender by referring to the sequence number of recently received packet.

If the sequence number of a segment recently received does not match with the sequence number the receiver was expecting, then it is discarded and NACK is sent back. If two segments arrive with the same sequence number, the TCP timestamp value is compared to make a decision.

18.7 Multiplexing

The technique to combine two or more data streams in one session is called Multiplexing. When a TCP client initializes a connection with Server, it always refers to a well-defined port number which indicates the application process. The client itself uses a randomly generated port number from private port number pools.

Using TCP Multiplexing, a client can communicate with a number of different application process in a single session. For example, a client requests a web page which in turn contains different types of data (HTTP, SMTP, FTP etc.) the TCP session timeout is increased and the session is kept open for longer time so that the three-way handshake overhead can be avoided.

This enables the client system to receive multiple connection over single virtual connection. These virtual connections are not good for Servers if the timeout is too long.

18.8 Congestion Control

When large amount of data is fed to system which is not capable of handling it, congestion occurs. TCP controls congestion by means of Window mechanism. TCP sets a window size telling the other end how much data segment to send. TCP may use three algorithms for congestion control:

- Additive increase, Multiplicative Decrease
- Slow Start
- Timeout React

18.9 Timer Management

TCP uses different types of timer to control and management various tasks:

Keep-alive timer:

- This timer is used to check the integrity and validity of a connection.
- When keep-alive time expires, the host sends a probe to check if the connection still exists.

Retransmission timer:

- This timer maintains stateful session of data sent.
- If the acknowledgement of sent data does not receive within the Retransmission time, the data segment is sent again.

Persist timer:

- TCP session can be paused by either host by sending Window Size 0.
- To resume the session a host needs to send Window Size with some larger value.

- If this segment never reaches the other end, both ends may wait for each other for infinite time.
- When the Persist timer expires, the host re-sends its window size to let the other end know.
- Persist Timer helps avoid deadlocks in communication.

Timed-Wait:

- After releasing a connection, either of the hosts waits for a Timed-Wait time to terminate the connection completely.
- This is in order to make sure that the other end has received the acknowledgement of its connection termination request.
- Timed-out can be a maximum of 240 seconds (4 minutes).

18.10 Crash Recovery

TCP is very reliable protocol. It provides sequence number to each of byte sent in segment. It provides the feedback mechanism i.e. when a host receives a packet, it is bound to ACK that packet having the next sequence number expected (if it is not the last segment).

When a TCP Server crashes mid-way communication and re-starts its process it sends TPDU broadcast to all its hosts. The hosts can then send the last data segment which was never unacknowledged and carry onwards.

19. Layer 4 – UDP

The User Datagram Protocol (UDP) is simplest Transport Layer communication protocol available of the TCP/IP protocol suite. It involves minimum amount of communication mechanism. UDP is said to be an unreliable transport protocol but it uses IP services which provides best effort delivery mechanism.

In UDP, the receiver does not generate an acknowledgement of packet received and in turn, the sender does not wait for any acknowledgement of packet sent. This shortcoming makes this protocol unreliable as well as easier on processing.

19.1 Requirement of UDP

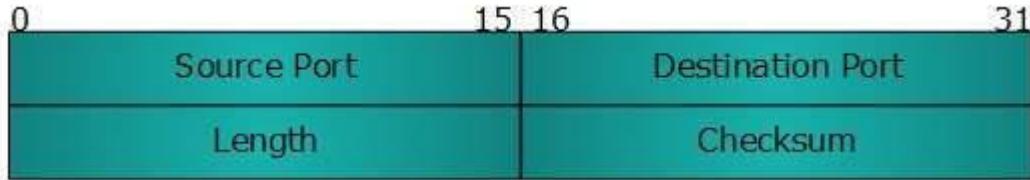
A question may arise, why do we need an unreliable protocol to transport the data? We deploy UDP where the acknowledgement packets share significant amount of bandwidth along with the actual data. For example, in case of video streaming, thousands of packets are forwarded towards its users. Acknowledging all the packets is troublesome and may contain huge amount of bandwidth wastage. The best delivery mechanism of underlying IP protocol ensures best efforts to deliver its packets, but even if some packets in video streaming get lost, the impact is not calamitous and can be ignored easily. Loss of few packets in video and voice traffic sometimes goes unnoticed.

19.2 Features

- UDP is used when acknowledgement of data does not hold any significance.
- UDP is good protocol for data flowing in one direction.
- UDP is simple and suitable for query based communications.
- UDP is not connection oriented.
- UDP does not provide congestion control mechanism.
- UDP does not guarantee ordered delivery of data.
- UDP is stateless.
- UDP is suitable protocol for streaming applications such as VoIP, multimedia streaming.

19.3 UDP Header

UDP header is as simple as its function.



UDP header contains four main parameters:

- **Source Port** - This 16 bits information is used to identify the source port of the packet.
- **Destination Port** - This 16 bits information, is used identify application level service on destination machine.
- **Length** - Length field specifies the entire length of UDP packet (including header). It is 16-bits field and minimum value is 8-byte, i.e. the size of UDP header itself.
- **Checksum** - This field stores the checksum value generated by the sender before sending. IPv4 has this field as optional so when checksum field does not contain any value it is made 0 and all its bits are set to zero.

19.4 UDP application

Here are few applications where UDP is used to transmit data:

- Domain Name Services
- Simple Network Management Protocol
- Trivial File Transfer Protocol
- Routing Information Protocol
- Kerberos

20. Layer 4 – TFTP

20.1 TFTP RRQ (Read Request) and TFTP WRQ (Write Request) messages

Mainly there are two types of TFTP requests.

- [TFTP RRQ \(Read Request\) message packet](#) is used to read a file from the root folder of the TFTP server.
- [TFTP WRQ \(Write Request\) message packet](#) is used to write a file to the root folder of the TFTP server.

[TFTP RRQ \(Read Request\)](#) and [TFTP WRQ \(Write Request\) messages](#) are also used to establish a TFTP connection from a [TFTP Client](#) to a [TFTP Server](#). Once the [TFTP RRQ message](#) or [TFTP WRQ message](#) is received at the TFTP Server, the data transfer is started either from the [TFTP Server](#) side or [TFTP Client](#) side.

20.2 TFTP DATA packet size

In TFTP, the file is divided into different blocks of data consisting of 512 bytes, except for the last block. The last block is identified by its size less than 512 bytes. The size of last block is between 0 and 511 bytes. In below example, we are trying to copy a file named "yamaha-rx100.txt" from the root folder of the [TFTP Server](#) to the [TFTP Client](#). The size of the file is 1.59 KB (1,630 bytes). Since the file size is 1,630 bytes, there must be four Blocks of [Data packets](#), from [TFTP Server](#) to [TFTP Client](#).

Please refer below table and then compare with the capture screenshot images provided at each steps below to understand the concept.

Sl. No.	Block Number	Block Size
1	Block Number 1	512 bytes
2	Block Number 2	512 bytes
3	Block Number 3	512 bytes
4	Block Number 4	94 bytes
Total		1630 bytes

To keep the protocol simple, TFTP uses [UDP](#) instead of [TCP](#) at [Transport Layer](#). UDP is not a reliable protocol. To provide basic level of reliability, TFTP uses [ACK messages](#) to confirm the receipt of blocks of data. When a block of data is received by [TFTP client](#) or [TFTP Server](#), a [TFTP ACK message](#) is sent back to acknowledge the receipt of the [Block of Data](#). If a Block of data is lost in transfer, TFTP Client will not send the acknowledgement message. Then, a timeout will happen at the TFTP Server, and the lost [TFTP DATA packet](#) is sent again.

20.3 TFTP port numbers

By default, **TFTP Server** is listening to UDP port 69, but you can make it listen any UDP **port number**, by changing the **port number** configuration at the TFTP Server. Initial **TFTP RRQ (Read Request)** and **TFTP WRQ (Write Request) message** packets are sent to UDP port 69. Further communications from **TFTP Server** back to the **TFTP Client** is from a random **port number** belongs to **Dynamic port number range**.

TFTP Client uses a random **port number** belongs to **Dynamic port number range**.

You can verify the UDP **ports** used by **TFTP Server** and **TFTP Client** at packet capture screenshots copied in each of the below steps. First image in each step shows a graphical representation of the direction of packet flow and the second image is Wireshark capture screenshot of the corresponding TFTP message.

20.4 How TFTP (Trivial File Transfer Protocol) works

Below example explains about **TFTP RRQ (Read Request) message** and how file copy from **TFTP Server** to **TFTP Client** works. **TFTP WRQ (Write Request) message** and file write process to the **TFTP Server** works similar to below example, but with reverse flow of **TFTP DATA packets** and **TFTP ACK packets**.

The IP addresses of the TFTP Client computer and the TFTP Server computer used in this example are 192.168.10.100 and 192.168.10.10 respectively.

Step 1 - The first step in coping a file from **TFTP Server** to **TFTP Client** is by sending a **TFTP RRQ (Read Request) message packet** to the TFTP server. By default, TFTP Server is listening to UDP **port** 69 and therefore the initial **TFTP RRQ (Read Request) message** is sent to UDP **port** 69.

The Opcode for **TFTP RRQ message** is 1 and **TFTP WRQ message** is 2. Mode (Type) of transfer for text file is netascii.

Step 2 - When a **TFTP RRQ (Read Request) message** is received to copy a file, TFTP server will reply back either with the first Block of data or with a **TFTP ERROR message**. Here in this case, everything is perfect and the reply is with first block of data (512 bytes).

Main point to note here is that further communications from **TFTP Server** to **TFTP Client** is not from UDP **port** 69. **TFTP Server** selects a dynamic UDP **port** number above 1023 for all further communications.

Step 3 - As discussed above, in TFTP, file transfer is made by slicing the file into blocks of data and sending blocks of data to the recipient. You can see that there are four blocks in this example, and they are sent in order from Block number 1 to Block number 4 to [TFTP Client](#). TFTP Client acknowledges each Block of data by replying with a [TFTP ACK message packet](#).

Once the first Block of data is received, the [TFTP Client](#) will send back the acknowledgement for the first Block of data (Block number 1).

Step 4 - Once the [TFTP ACK message](#) for first Block of Data is received from the [TFTP Client](#), the [TFTP Server](#) sends the second Block of Data.

Step 5 - Once the second Block of data is received, the [TFTP Client](#) will send back the acknowledgement for the second Block of data.

Step 6 - Once the [TFTP ACK message](#) for the second Block of Data received, the [TFTP Server](#) sends the third Block of data.

Step 7 - Once the third Block of data is received, the **TFTP Client** will send back the acknowledgement for the third Block of data.

Step 8 - In this example, the total size of the file being transferred is 1,630 bytes. 1,536 bytes of data is already transferred in previous three blocks of Data transfer (512×3). Remaining is 94 bytes of data to transfer to the **TFTP Client**. Once the **TFTP ACK message** for the third Block of Data received, the TFTP Server sends the last Block of data, consisting of remaining 94 bytes of data.

Again, note that in TFTP, the last Block of Data is identified by its file size, which is less than 512 bytes (in this example 94 bytes).

Step 9 - Once the last Block of data is received, the **TFTP Client** will send back the acknowledgement for the fourth Block of data.

The TFTP file transfer is completed now.

21. Layer 7 – SOAP

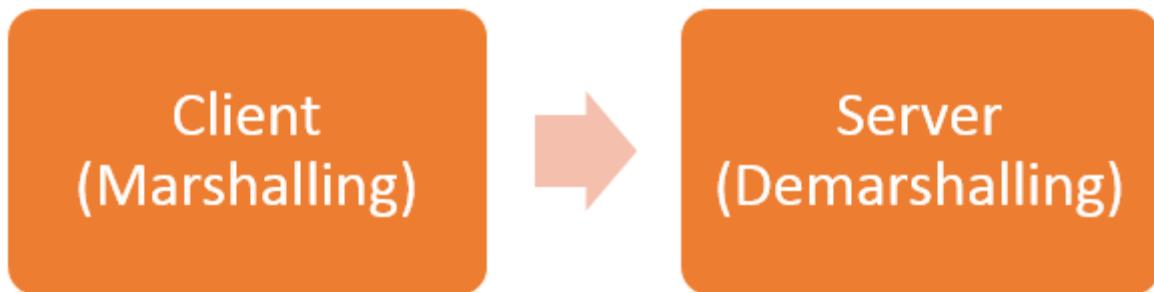
SOAP is an XML-based protocol for accessing web services over HTTP.

SOAP was developed as an intermediate language so that applications built on various programming languages could talk easily to each other and avoid the extreme development effort.

One of the methods used to combat this complexity is to use XML (Extensible Markup Language) as the intermediate language for exchanging data between applications.

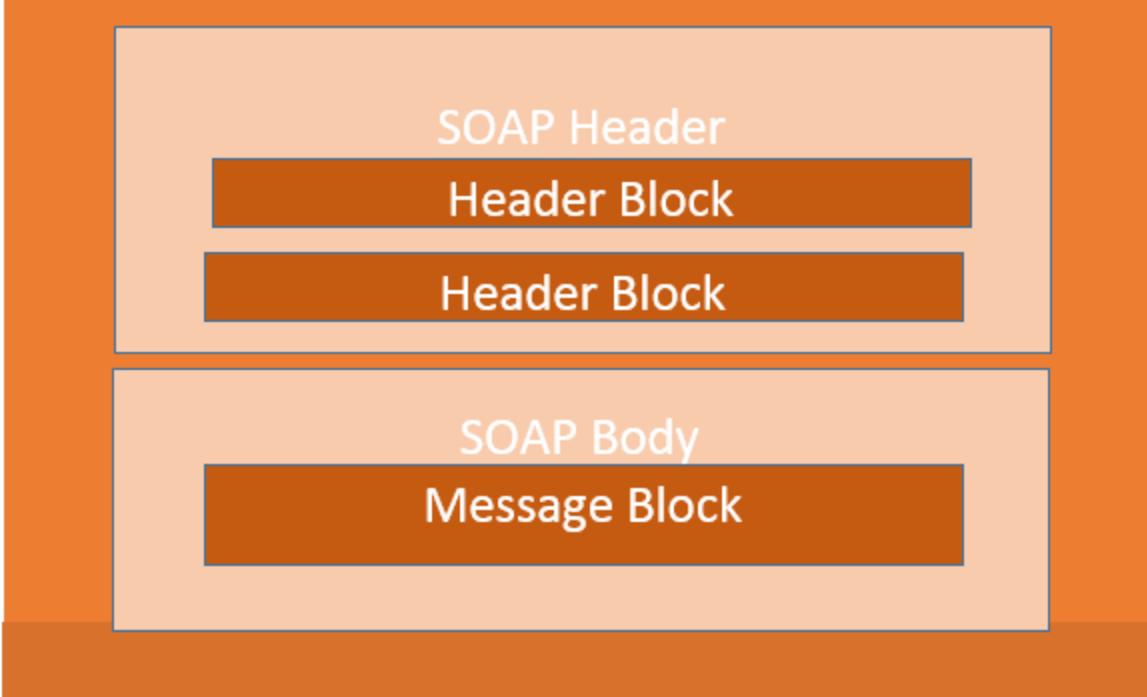
Every programming language can understand the XML markup language. Hence, XML was used as the underlying medium for data exchange.

To overcome all of the limitations cited above, SOAP would then use the below communication model



1. The client would format the information regarding the procedure call and any arguments into a SOAP message and sends it to the server as part of an HTTP request. This process of encapsulating the data into a SOAP message was known as **Marshalling**.
2. The server would then unwrap the message sent by the client, see what the client requested for and then send the appropriate response back to the client as a SOAP message. The practice of unwrapping a request sent by the client is known as **Demarshalling**.

SOAP Envelope



The SOAP message is nothing but a mere XML document which has the below components.

- An Envelope element that identifies the XML document as a SOAP message – This is the containing part of the SOAP message and is used to encapsulate all the details in the SOAP message. This is the root element in the SOAP message.
- A Header element that contains header information – The header element can contain information such as authentication credentials which can be used by the calling application. It can also contain the definition of complex types which could be used in the SOAP message. By default, the SOAP message can contain parameters which could be of

simple types such as strings and numbers, but can also be a complex object type.

- The Envelope element
- The header element and
- The body element
- The Fault element (Optional)

Let's look at an example below of a simple SOAP message and see what element actually does.



SOAP Message Structure

1. As seen from the above SOAP message, the first part of the SOAP message is the envelope element which is used to encapsulate the entire SOAP message.
2. The next element is the SOAP body which contains the details of the actual message.
3. Our message contains a web service which has the name of "Guru99WebService".
4. The "Guru99Webservice" accepts a parameter of the type 'int' and has the name of TutorialID.

Now, the above SOAP message will be passed between the web service and the client application.

You can see how useful the above information is to the client application. The SOAP message tells the client application what is the name of the Web service, and also what parameters it expects and also what is the type of each parameter which is taken by the web service.

1. **<faultCode>**— This is the code that designates the code of the error. The fault code can be either of any below values
 1. SOAP-ENV:VersionMismatch – This is when an invalid namespace for the SOAP Envelope element is encountered.
 2. SOAP-ENV:MustUnderstand – An immediate child element of the Header element, with the mustUnderstand attribute set to “1”, was not understood.
 3. SOAP-ENV:Client – The message was incorrectly formed or contained incorrect information.
 4. SOAP-ENV:Server – There was a problem with the server, so the message could not proceed.
2. **<faultString>** – This is the text message which gives a detailed description of the error.
3. **<faultActor> (Optional)**— This is a text string which indicates who caused the fault.
4. **<detail>(Optional)** – This is the element for application-specific error messages. So the application could have a specific error message for different business logic scenarios.

22. Layer 7 – SSDP

SSDP (Simple Service Discovery Protocol) is a network protocol used in small networks, including home networks, to advertise and discover network services primarily supported by the Universal Plug-and-Play (UPnP) architecture. SSDP is an HTTPU-based textual protocol that uses XML. It exchanges messages using [UDP](#) datagrams.

On devices and PCs that support SSDP, this feature can be enabled, disabled, or paused. When SSDP is enabled, devices communicate information about themselves and the services they provide to any other UPnP client. Using SSDP, computers connected to the network also provide information about available services.

In its most general form, the connection of a new device looks like this. To find out which devices are already present on the network, a device added to it with SSDP enabled sends a search request to other devices to the reserved address and port (239.255.255.250:1900), using fan-out or multicasting. In the request, the device specifies a template or target corresponding to its type. In response to the request, each of the devices on the network that support SSDP at the moment sends a UDP message with information about itself to the source IP address and port from which the request was sent.

23. Layer 7 – DHCP

DHCP :Dynamic Host Configuration Protocol : Application layer protocol that automates the configuration process of hosts on a network.

Dynamic IP allocation to hosts connecting to it and requesting

Automatic allocation : A range of IP addresses is set aside for assignment purposes

Fixed allocation:Requires a manually specified list of MAC/IP addresses

NTP : Network Time Protocol – used to keep all computers in network synchronized in time.

DHCP discovery: the process by which a client configured to use DHCP attempts to get network configuration information.

Client sends DHCP discover message to find DHCP server

DHCP listens on UDP 67, all DHCP requests are sent from port 68 to a broadcast destination IP

Src ip : 0.0.0.0 dst ip 255.255.255.255

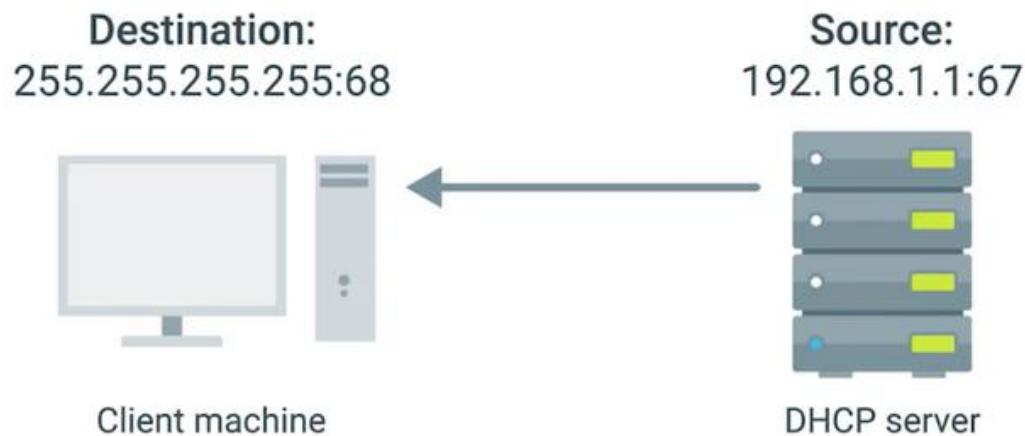
Every node of the local area network gets the request

DHCPDISCOVER



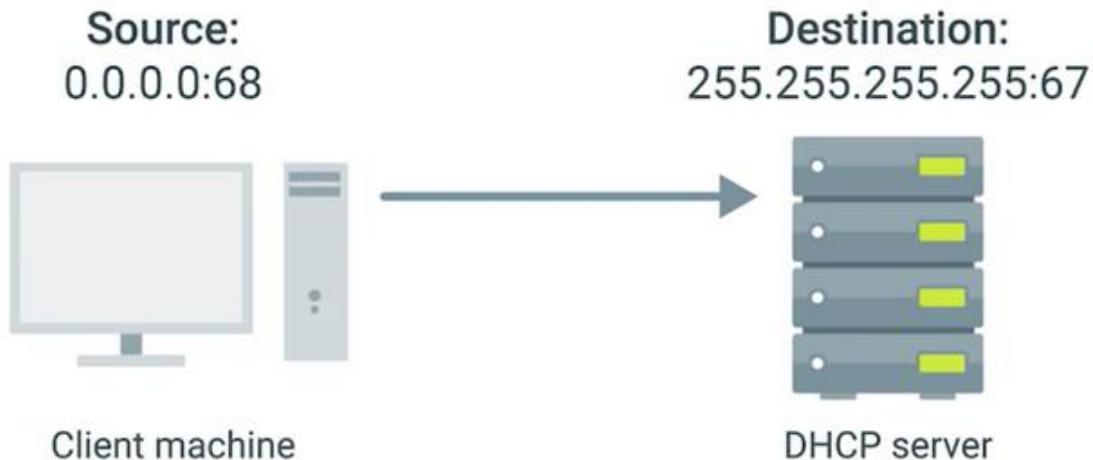
Response is DHCP offer message

DHCPOFFER



Offer is also broadcast, received by every host, but the response will have information of client MAC address
DHCP client can respond to the offer with DHCP request message

DHCPREQUEST



DHCP ACK is sent back and IP assignment is done

DHCP Lease

DHCP uses transaction IDs

1hr limit

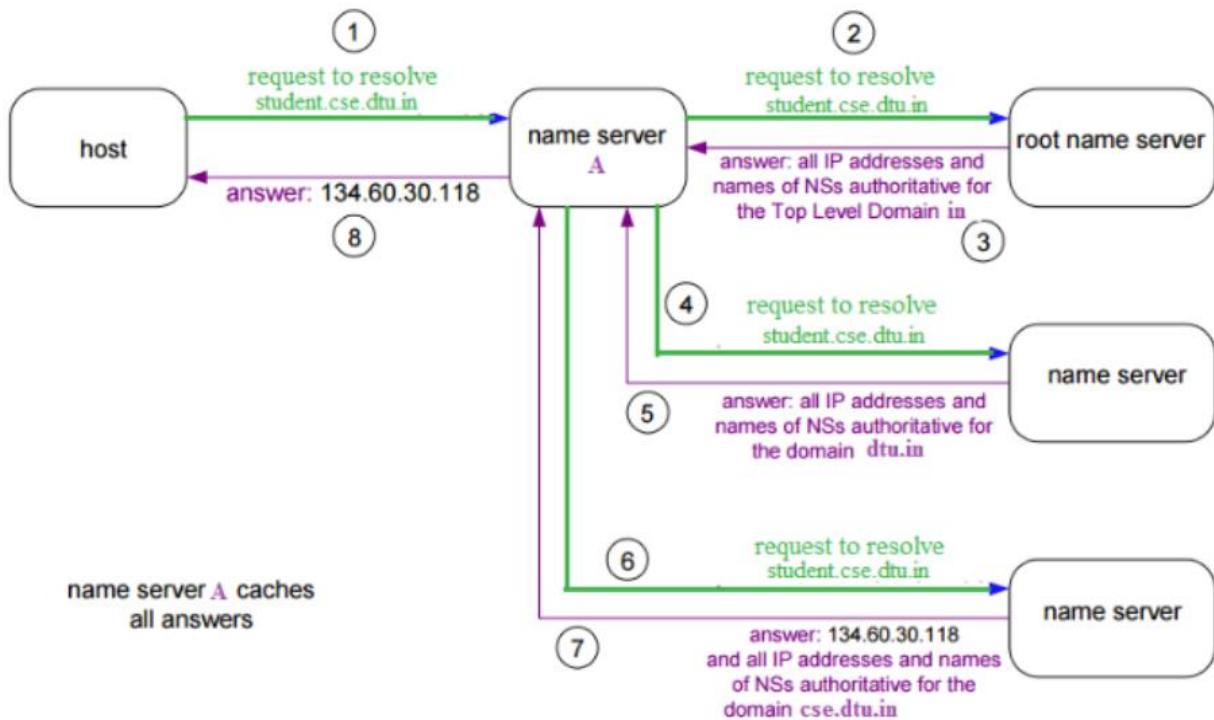
Network Address Translation (NAT) is a technology that allows the gateway usually a router or firewall to rewrite the source IP of an outgoing IP datagram while retaining the original IP in order to rewrite it into the response.

IP masquerading – a firewall having NAT changing the src ip of packet from host to itself (one to many NAT)

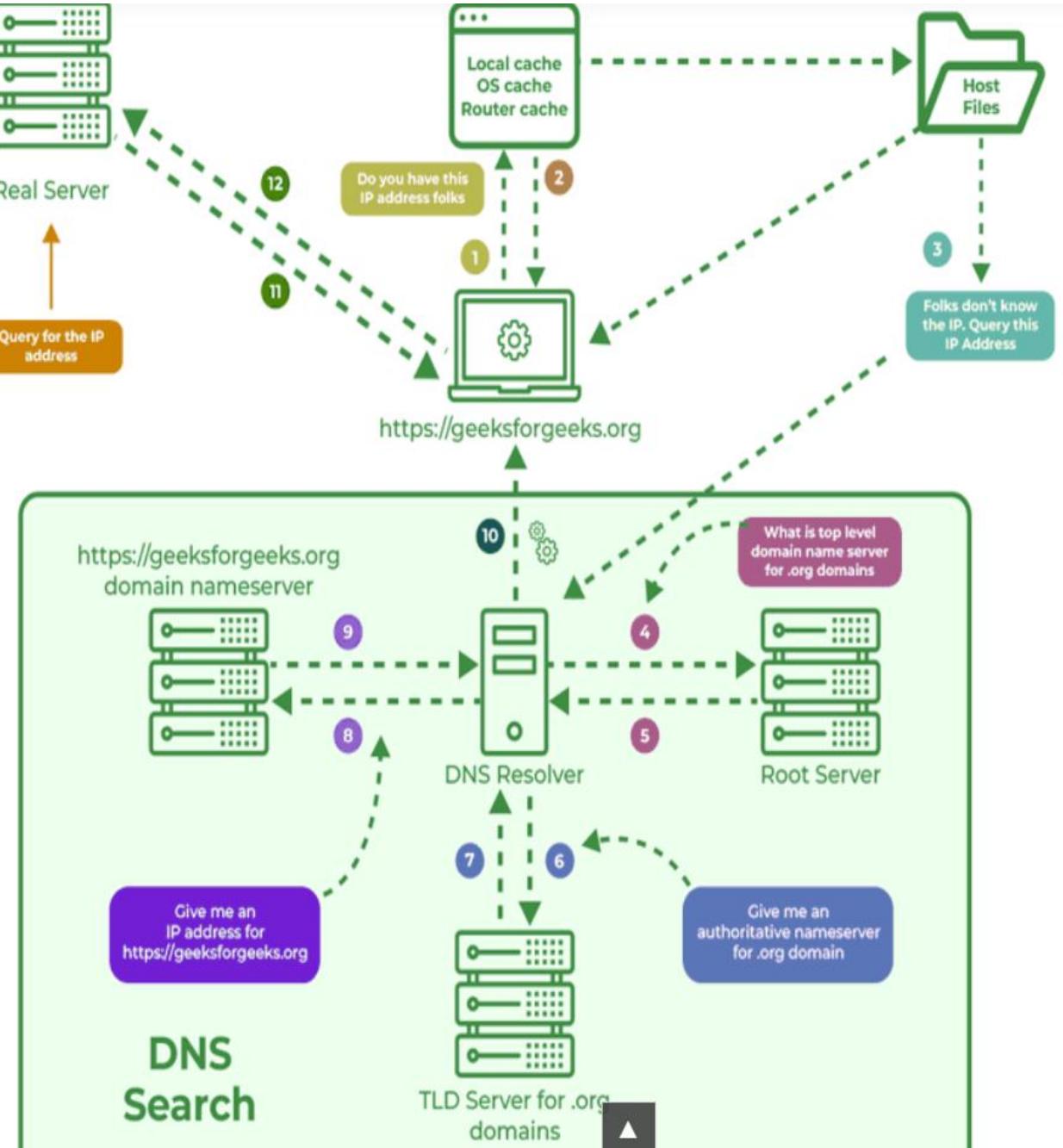
Port preservation : src port used by host is the same used by router

Port forwarding : where specific destination ports can be configured to always be delivered to specific nodes.

24. Layer 7 – DNS



Domain Name Server



Actually in a URL there is a final .

www.website.com, (last dot) is the root server out of 13 (FQDN -> Fully qualified domain name)

Domain naming server or system

Root Server - > top level domain server

.com -is TLD

SLD -> second level domain server (www.learning.linkedin.com)

Domain names are reserved and registered by domain registrar(ICANN register)

Subdomain(learning)

Forward(domain to ip) and reverse lookup(ip to domain)

Authoritative name server is the ip which actually holds the domain or service we are looking for

DNS lookup happens with DNS resolver (it could be on your PC or LAN or internet)

DNS resolver returns the information in its cache (non recursive query)

DNS response is marked as non-authoritative

DNS resolver does a recursive lookup if it does not have the information (authoritative)
Recursive name servers only know about root name servers
Dig is a linux command for looking up DNS records
A record – ipv4 address
Quad A record – IPV6 address
TTL – amount of time dns resolver will store this record in cache
CNAME – short canonical form for the actual website address we are typing in -> FQDN
MX – mail exchange records (preference – lower the better among many)
SPF – sender policy framework or TXT record – authenticate incoming/outgoing mail (human readable notes)
NS – name server - has all name servers included
SOA – start of authority – name, serial, refresh, expire , ttl (primary name server)
PTR – opposite of A and AAAA (reverse lookup)
SRV – Service – some additional info required for instant messaging service

DNS forwarding – forwarding to external DNS resolvers (public)
DNS Security – spoofing – (attacker changes the domain name)
DNS hijacking – redirect to wrong dns server
DNS Cache poisoning – alter dns records in server
Change DNS port settings, redundant DNS Server
DNSSEC- checks the authenticity of DNS queries using digital signatures, firewall

Cat /etc/resolv.conf – dns server(name server)
/etc/hosts
Dns lookup – host
Nslookup
Dig
Resolvectl

DNS – Domain Name System – mapping between domain names and their respective IPs(name resolution)
Domain Names : Something that can be resolved by DNS
For specific HW – there is MAC address
For every node in the N/W – IP address, Subnet Mask, Gateway Address and DNS server
Caching Name Servers – provided by ISP(Store known domain name lookups for a certain amount of time) - caching
Recursive Name Servers – provided by ISP (Store known domain name lookups for a certain amount of time) – performs full DNS resolution requests
TTL is the time to live – configured by owner of domain name for how long a name server is allowed to cache an entry before doing a full resolution again.

Anycast: route traffic to different destinations depending on location, congestion or link health.
Root Name Servers – 13 authorities that provide root name lookup as a service. Root servers respond back to the DNS requests with a TLD name server that has to be queried

TLD Name Servers – Top level domain, represents top of the hierarchical DNS name resolution system
For each TLD there is a TLD name server

www.google.com (com is TLD)

TLD responds with what authoritative name server to contact (anycast)

Authoritative Name Servers – actual organization that holds the site – actual IP is returned from here

DNS (port 53) uses UDP (request and response for dns lookup can fit in a single datagram of udp)
DNS application has error recovery – will query again if response not received
If DNS response is too big to fit in a UDP datagram, response is given as too large, then DNS over TCP is used.

Resource Record Types:

A Record: Used to point a certain domain name at a certain IPv4 address

Single domain name can have 1 or more A record (DNS round robin)

Quad A record – Ipv6 instead of Ipv4 (round robin is used for load balancing of traffic rate to a web site handled by multiple A records)

CNAME record: Used to redirect traffic from one domain to another domain name (Microsoft.com to www.microsoft.com)

By setting canonical names, maintainability is easy), if ip changes just one A record change is done.

MX Record: Mail exchange record – resource record to deliver mail to correct server

SRV (service record) – define location of various specific services – calendar and scheduling

(TX)Text record: some descriptive text for a domain

Administration and definition of TLDs is done by ICANN Internet Corporation for Assigned Names and Numbers

.com – Top level domain

Google – domain (where control moves from TLD name server to authoritative name server)

Www – sub domain

Fully Qualified Domain Name

A registrar is someone who can sell unregistered domain names – has agreement with ICANN

DNS can support up to 127 levels of domain in total for a single FQDN

Each individual section can only be 63 characters long

Complete FQDN is 255 characters long.

DNS Zones – allows for easier control over multiple levels of domain.

Zone files : Simple configuration files that declare all resource records for a particular zone.

Contains SOA (start of authority – declares the zone and the name of the name server that it is authoritative for), resource record declaration

NS records - other name servers responsible for this zone.

Reverse lookup zone files: DNS resolvers ask for an IP and get the FQDN with it returned

25. Layer 7 – HTTP

Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other. HTTP specification specifies how clients' request data will be constructed and sent to the server, and how the servers respond to these requests.

- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnects the connection. So client and server know about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.
- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.
- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

Version HTTP uses a <major>.<minor> numbering scheme to indicate versions of the protocol. The version of an HTTP message is indicated by an HTTP-Version field in the first line. Here is the general syntax of specifying HTTP version number:

```
HTTP-Version = "HTTP" "/" 1*DIGIT "." 1*DIGIT
```

Uniform Resource Identifiers (URI) are simply formatted, case-insensitive string containing name, location, etc. to identify a resource, for example, a website, a web service, etc. A general syntax of URI used for HTTP is as follows:

```
URI = "http://" // host [ ":" port ] [ abs_path [ "?" query ] ]
```

Here if the **port** is empty or not given, port 80 is assumed for HTTP and an empty **abs_path** is equivalent to an **abs_path** of "/". The characters other than those in the **reserved** and **unsafe** sets are equivalent to their "% HEX HEX" encoding

Date and Time All HTTP date/time stamps MUST be represented in Greenwich Mean Time (GMT), without exception.

Character Set We use character sets to specify the character sets that the client prefers.

A **content encoding** value indicates that an encoding algorithm has been used to encode the content before passing it over the network

HTTP uses Internet Media Types in the **Content-Type** and **Accept** header fields in order to provide open and extensible data typing and type negotiation.

GET /hello.htm HTTP/1.1 (This is Request-Line sent by the client)

HTTP/1.1 200 OK (This is Status-Line sent by the server)

- **General-header:** These header fields have general applicability for both request and response messages.
- **Request-header:** These header fields have applicability only for request messages.

- **Response-header:** These header fields have applicability only for response messages.
- **Entity-header:** These header fields define meta information about the entity-body or, if no body is present, about the resource identified by the request.
- A message body is the one which carries the actual HTTP request data (including form data and uploaded, etc.) and HTTP response data from the server (including files, images, etc.). Shown below is the simple content of a message body:

```

• <html>
•   <body>
•
•   <h1>Hello, World!</h1>
•
•   </body>
• </html>
```

GET

The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.

2

HEAD

Same as GET, but it transfers the status line and the header section only.

3

POST

A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.

	PUT Replaces all the current representations of the target resource with the uploaded content.
4	DELETE Removes all the current representations of the target resource given by URI.
5	CONNECT Establishes a tunnel to the server identified by a given URI.
6	OPTIONS Describe the communication options for the target resource.
7	TRACE Performs a message loop back test along with the path to the target resource.
8	1xx: Informational It means the request was received and the process is continuing.
1	2xx: Success It means the action was successfully received, understood, and accepted.
2	3xx: Redirection It means further action must be taken in order to complete the request.
3	

4

4xx: Client Error

It means the request contains incorrect syntax or cannot be fulfilled.

5

5xx: Server Error

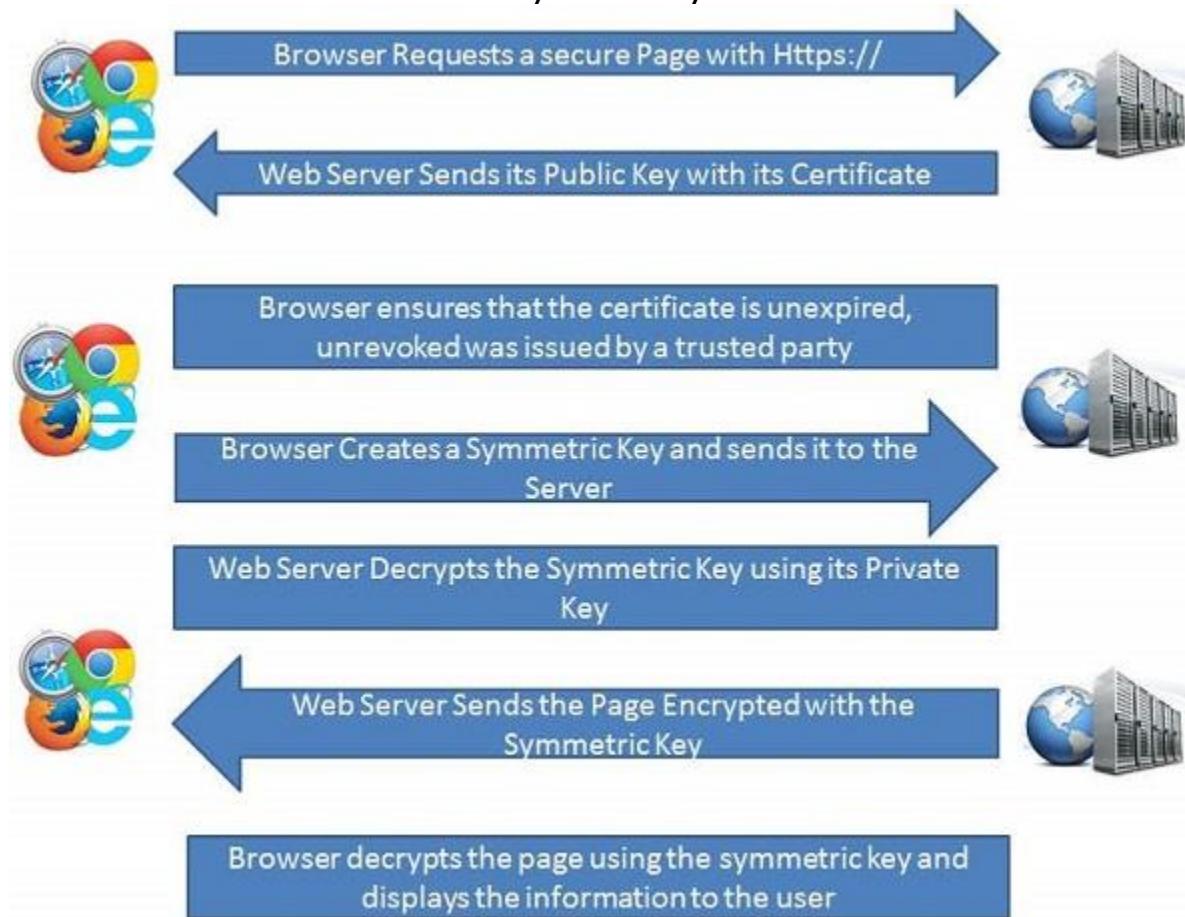
It means the server failed to fulfill an apparently valid request.

HTTPS (Hypertext Transfer Protocol over Secure Socket Layer) or HTTP over SSL is a web protocol developed by Netscape. It is not a protocol but it is just the result of layering the HTTP on top of SSL/TLS (Secure Socket Layer/Transport Layer Security).

In short, HTTPS = HTTP + SSL

26. Layer 7 – HTTPS - SSL?

- Public key and signed certificates are required for the server in HTTPS Protocol.
- Client requests for the https:// page
- When using an https connection, the server responds to the initial connection by offering a list of encryption methods the webserver supports.
- In response, the client selects a connection method, and the client and server exchange certificates to authenticate their identities.
- After this is done, both webserver and client exchange the encrypted information after ensuring that both are using the same key, and the connection is closed.
- For hosting https connections, a server must have a public key certificate, which embeds key information with a verification of the key owner's identity.
- Almost all certificates are verified by a third party so that clients are assured that the key is always secure.



As you learned in the previous chapter, https uses SSL protocol to secure the communication by transferring encrypted data. Before going deeper, learn how SSL works.

SSL fundamentally works with the following concepts:

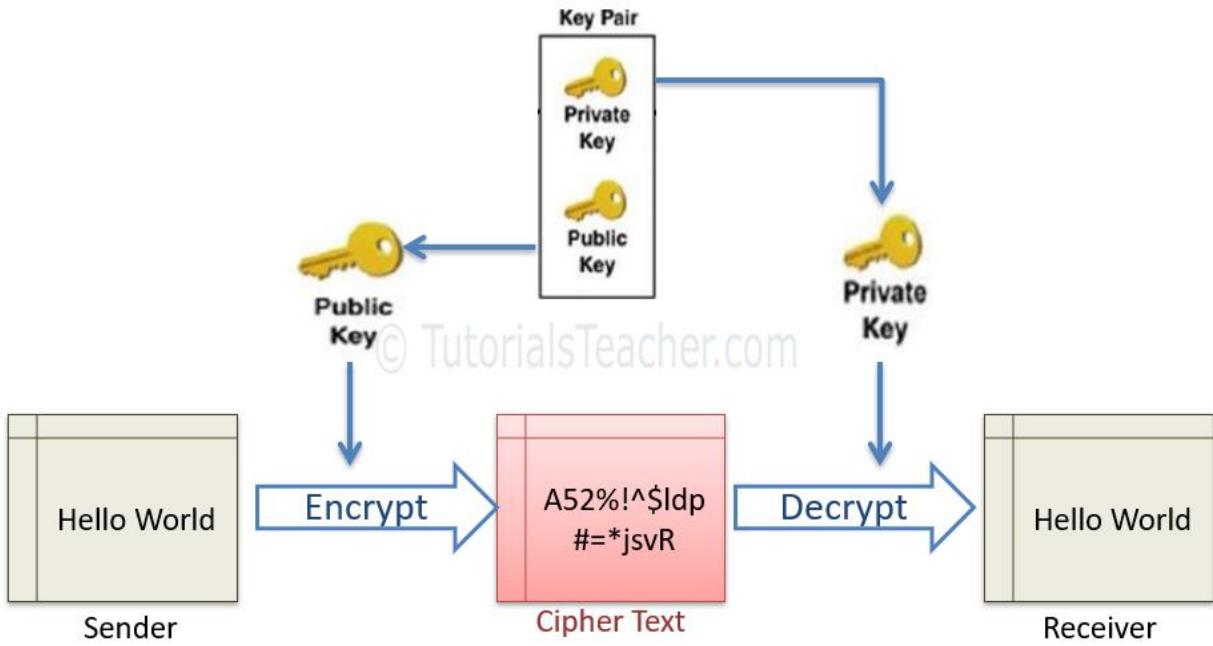
1. Asymmetric Cryptography
2. Symmetric Cryptography

26.1 Asymmetric Cryptography

Asymmetric cryptography (also known as Asymmetric Encryption or Public Key Cryptography) uses a mathematically-related key pair to encrypt and decrypt data. In a key pair, one key is shared with anyone who is interested in a communication. This is called **Public Key**. The other key in the key pair is kept secret and is called **Private Key**.

Public and private keys are mathematical related and were created using cryptographic algorithms which are based on mathematical problems termed one-way functions. These keys are used to encrypts or decrypts the data.

In the asymmetric cryptography, the sender encrypt data with the receiver's public key and send it to the receiver. The receiver decrypts it using the related private key.



Asymmetric Cryptography

SSL uses asymmetric cryptography to initiate the communication which is known as SSL handshake. Most commonly used asymmetric key encryption algorithms include EIGamal, RSA, DSA, Elliptic curve techniques and PKCS.

26.2 Symmetric Cryptography

In the symmetric cryptography, there is only one key which encrypts and decrypts the data. Both sender and receiver should have this key, which is only known to them.

Symmetric Cryptography

SSL uses symmetric cryptography using the session key after the initial handshake is done. The most widely used symmetric algorithms are AES-128, AES-192 and AES-256.

26.3 Data Transfer over SSL

SSL protocol uses asymmetric and symmetric cryptography to transfer data securely. The following figure illustrates the steps of SSL communication:

SSL Communication

As you can see in the above figure, SSL communication between the browser and the web server (or any other two systems) is mainly divided into two steps: the SSL handshake and the actual data transfer.



SSL Handshake

The communication over SSL always begins with the SSL handshake. The SSL handshake is an asymmetric cryptography which allows the browser

to verify the web server, get the public key and establish a secure connection before the beginning of the actual data transfer.

The following figure illustrates the steps involved in the SSL handshake:

SSL Handshake

Let's understand the above steps:

1. The client sends a "client hello" message. This includes the client's SSL version number, cipher settings, session-specific data and other information that the server needs to communicate with the client using SSL.
2. The server responds with a "server hello" message. This includes the server's SSL version number, cipher settings, session-specific data, an SSL certificate with a public key and other information that the client needs to communicate with the server over SSL.
3. The client verifies the server's SSL certificate from CA (Certificate Authority) and authenticates the server. If the authentication fails, then the client refuses the SSL connection and throws an exception. If the authentication succeeds, then proceed to step 4.
4. The client creates a session key, encrypts it with the server's public key and sends it to the server. If the server has requested client authentication (mostly in server to server communication), then the client sends his own certificate to the server.
5. The server decrypts the session key with its private key and sends the acknowledgement to the client encrypted with the session key.

Thus, at the end of the SSL handshake, both the client and the server have a valid session key which they will use to encrypt or decrypt actual data. The public key and the private key will not be used any more after this.

26.4 Actual Data Transfer

The client and the server now use a shared session key to encrypt and decrypt actual data and transfer it. This is done using the same session key at both ends and so, it is a symmetric cryptography. The actual SSL data transfer uses symmetric cryptography because it is easy and takes less CPU consumption compared with the asymmetric cryptography.

Thus, SSL fundamentally works using asymmetric cryptography and symmetric cryptography. There are certain infrastructures involved in achieving SSL communication in real life, which are called Public Key Infrastructure.

26.5 Public Key Infrastructure

The [public key infrastructure \(PKI\)](#) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public key encryption.

PKI includes the following elements:

- Certificate Authority: The authority that authenticates the identity of individuals, computers and other entities.
- Registration Authority: A subordinate CA that issues a certificate on the behalf of root CA for specific uses.
- SSL Certificate: The Data file that includes the public key and other information.
- Certificate Management System: The system which stores, validates and revokes certificates.
- **Choose a Certificate Authority (CA):** You can choose your CA from where you want to buy an SSL certificate. There are many CAs such as [Comodo](#), [DigiCert](#), [RapidSSL](#), [GeoTrust](#), [thawte](#), [Certum](#) etc. You may choose the CA based on your budget and the features you need to manage the certificate. There are resellers who provides cheap SSL certificates from these CAs. Visit <https://www.thesslstore.com> to buy cheap certificates from reputed CAs at one place.
- **Select the certificate you need:** Once you select a CA, you can choose the certificate you require for your website based on the validation method and the number of web sites you want to secure.
- **Purchase the certificate:** Once you choose the certificate you require, make the payment to proceed. For some CA, this step comes after submitting a CSR.
- **Generate and submit a CSR (Certificate Signing Request) to the CA:** You need to generate a CSR from your web server and submit it to the CA. Learn more about what a CSR is and how to generate it in the next chapter.

- **Download the SSL certificate (after successful validation):** After submitting a CSR, the CA will take some time for validating your information. This may vary based on the validation certificate you purchase. For a DV certificate, it should be quick. But, it will take a little longer to validate for an OV and an EV certificates. Once the CA successfully validates your information, you will get an email containing the certificate or you can download it from your account on the CA's website.
- **Install an SSL certificate on your web server:** Once you get your SSL certificate, you need to install it on the web server from where you generated your CSR. The installation process depends on the OS of your server.

A CSR includes a public key generated by the web server where you are going to install an SSL certificate. The web server generates a key pair, a public key and a private key when the CSR is generated. It includes a public key in the CSR and also stores a private key secretly in the file system. This private key will be used when installing a certificate on the web server.

27. Layer 7 – SMTP

Email is emerging as one of the most valuable services on the internet today. Most internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas **POP (post office protocol)** or **IMAP (internet message access protocol)** is used to retrieve those emails at the receiver's side.

SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is an always-on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection through port 25. After successfully establishing a TCP connection the client process sends the mail instantly.

SMTP



Use Case :- *Sending/Receiving Emails*



EE

SMT

The SMTP model is of two types:

- End-to-End Method
- Store-and-Forward Method

The end-to-end model is used to communicate between different organizations whereas the store and forward method is used within an organization. An SMTP client who wants to send the mail will contact the destination's host SMTP directly, to send the mail to the destination. The SMTP server will keep the mail to itself until it is successfully copied to the receiver's SMTP.

The client SMTP is the one that initiates the session so let us call it the client-SMTP and the server SMTP is the one that responds to the session request so let us call it receiver-SMTP. The client-SMTP will start the session and the receiver SMTP will respond to the request.

27.1 SMTP Protocol

The SMTP model is of two types:

- End-to-End Method
- Store-and-Forward Method

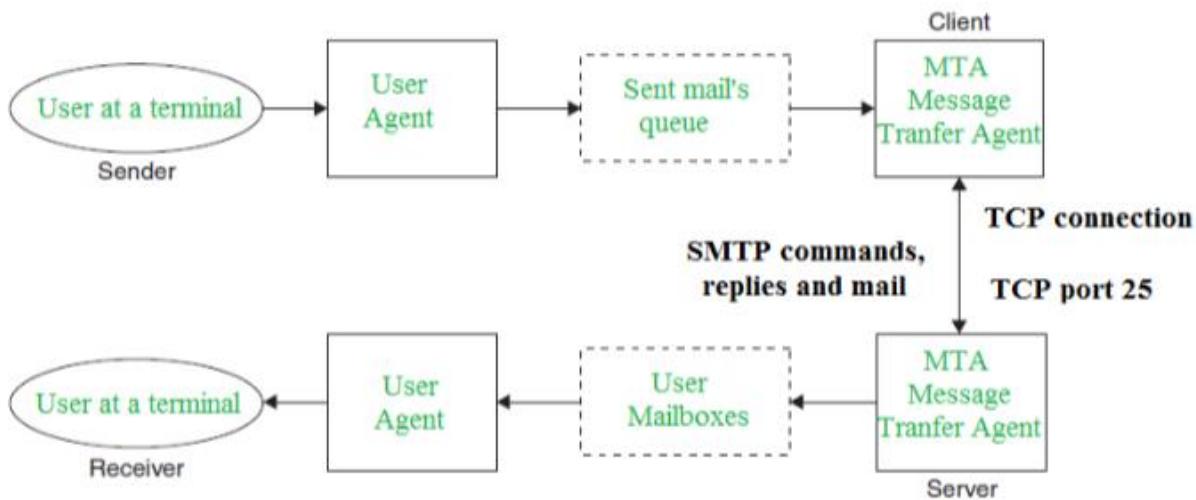
The end-to-end model is used to communicate between different organizations whereas the store and forward method is used within an organization. An SMTP client who wants to send the mail will contact the destination's host SMTP directly, to send the mail to the destination. The SMTP server will keep the mail to itself until it is successfully copied to the receiver's **SMTP**.

The client SMTP is the one that initiates the session so let us call it the client-SMTP and the server SMTP is the one that responds to the session request so let us call it receiver-SMTP. The client-SMTP will start the session and the receiver SMTP will respond to the request.

- **Mail User Agent (MUA):** It is a computer application that helps you in sending and retrieving mail. It is responsible for creating email messages for transfer to the mail transfer agent(MTA).
- **Mail Submission Agent (MSA):** It is a computer program that receives mail from a Mail User Agent(MUA) and interacts with the Mail Transfer Agent(MTA) for the transfer of the mail.
- **Mail Transfer Agent (MTA):** It is software that has the work to transfer mail from one system to another with the help of SMTP.
- **Mail Delivery Agent (MDA):** A mail Delivery agent or Local Delivery Agent is basically a system that helps in the delivery of mail to the local system.

27.2 Model of SMTP System

In the SMTP model user deals with the user agent (UA), for example, Microsoft Outlook, Netscape, Mozilla, etc. To exchange the mail using TCP, MTA is used. The user sending the mail doesn't have to deal with MTA as it is the responsibility of the system admin to set up a local MTA. The MTA maintains a small queue of mail so that it can schedule repeat delivery of mail in case the receiver is not available. The MTA delivers the mail to the mailboxes and the information can later be downloaded by the user agents.



SMTP Model

27.3 Components of SMTP

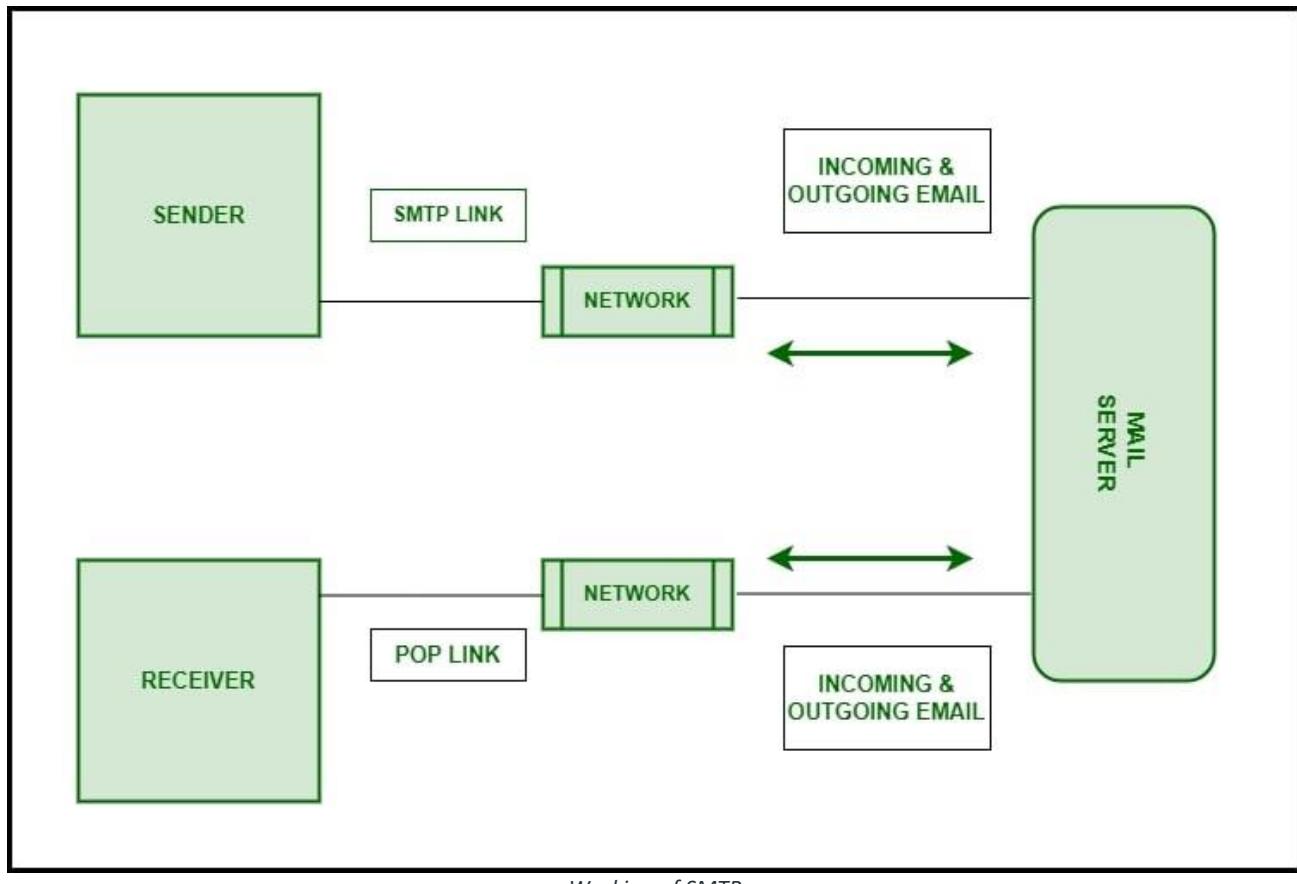
- **Mail User Agent (MUA):** It is a computer application that helps you in sending and retrieving mail. It is responsible for creating email messages for transfer to the mail transfer agent(MTA).
- **Mail Submission Agent (MSA):** It is a computer program that receives mail from a Mail User Agent(MUA) and interacts with the Mail Transfer Agent(MTA) for the transfer of the mail.
- **Mail Transfer Agent (MTA):** It is software that has the work to transfer mail from one system to another with the help of SMTP.
- **Mail Delivery Agent (MDA):** A mail Delivery agent or Local Delivery Agent is basically a system that helps in the delivery of mail to the local system.

27.4 Working of SMTP

Below mentioned are the steps of the working of SMTP [Simple Mail Transfer Protocol].

- **Communication between the sender and the receiver:** The sender's user agent prepares the message and sends it to the MTA. The MTA's responsibility is to transfer the mail across the network to the receiver's MTA. To send mail, a system must have a client MTA, and to receive mail, a system must have a server MTA.
- **Sending Emails:** Mail is sent by a series of request and response messages between the client and the server. The message which is sent across consists of a header and a body. A null line is used to terminate the mail header and everything after the null line is considered the body of the message, which is a sequence of ASCII characters. The message body contains the actual information read by the receipt.

- **Receiving Emails:** The user agent on the server-side checks the mailboxes at a particular time of intervals. If any information is received, it informs the user about the mail. When the user tries to read the mail it displays a list of emails with a short description of each mail in the mailbox. By selecting any of the mail users can view its contents on the terminal.



Working of SMTP

1.	HELO	HELO<SP><domain><CRLF>	It provides the identification of the sender i.e. the host name.	Mandatory
2.	MAIL	MAIL<SP>FROM : <reverse-path><CRLF>	It specifies the originator of the mail.	Mandatory
3.	RCPT	RCPT<SP>TO : <forward-path><CRLF>	It specifies the recipient of mail.	Mandatory

4.	DATA	DATA<CRLF>	It specifies the beginning of the mail.	Mandatory
5.	QUIT	QUIT<CRLF>	It closes the TCP connection.	Mandatory
6.	RSET	RSET<CRLF>	It aborts the current mail transaction but the TCP connection remains open.	Highly recommended
7.	VRFY	VRFY<SP><string><CRLF>	It is use to confirm or verify the user name.	Highly recommended
8.	NOOP	NOOP<CRLF>	No operation	Highly recommended
9.	TURN	TURN<CRLF>	It reverses the role of sender and receiver.	Seldom used
10.	EXPN	EXPN<SP><string><CRLF>	It specifies the mailing list to be expanded.	Seldom used
11.	HELP	HELP<SP><string><CRLF>	It send some specific documentation to the system.	Seldom used
12.	SEND	SEND<SP>FROM : <reverse-path><CRLF>	It send mail to the terminal.	Seldom used
13.	SOML	SOML<SP>FROM : <reverse-path><CRLF>	It send mail to the terminal if possible; otherwise to mailbox.	Seldom used
14.	SAML	SAML<SP>FROM : <reverse-path><CRLF>	It send mail to the terminal and mailbox.	Seldom use

28. Layer 7 – POP3

Today email is the most widely used platform for personal and professional business communication. Email protocol is defined as a set of rules for the secure transmission of emails over the Internet. SMTP, IMAP, POP, and POP3 are some of the email protocols. Each type of protocol used has a specific mechanism. The below article covers in detail the POP3 protocol.

28.1 What is POP?

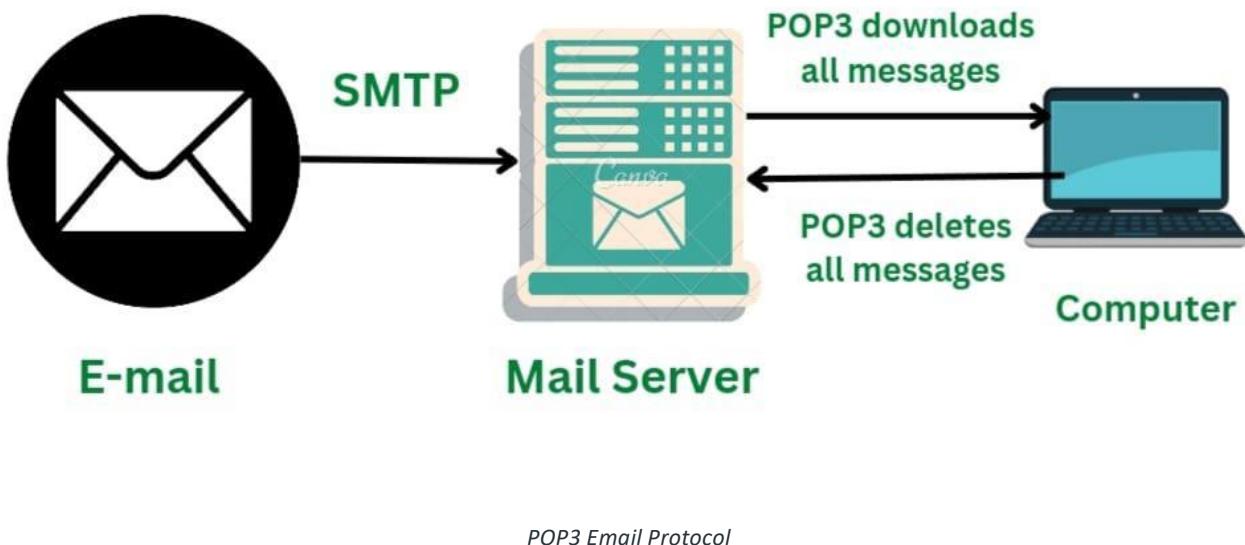
POP stands for Post Office Protocol. The [POP protocol](#) was published in the year 1984. POP has been updated two times namely “POP2” and “POP3”. The POP protocol is an Internet Standard Protocol that works on the application layer. It is used to get an access email from the mail server. The need for POP mainly arises when the user or client does not have a continuous internet connection and what's to receive email messages. The Pop client makes use of POP to pull email messages from the POP server. POP3 is the updated version of POP.

28.2 What is POP3?

POP 3 stands for Post Office Protocol Version 3. POP3 protocol is used to provide access to the mail inbox that is stored in the email server. POP3 protocol can download and delete messages. Once the POP3 client has established a connection with the mail server it can easily retrieve all the messages from the server. The user can access the messages locally even if the user is offline. Every time the client needs to check manually for new messages as POP3 Protocol provides the feature of real-time synchronization. Various email applications such as Microsoft Outlook, Apple Mail, Gmail supports POP3 protocol.

When a message is sent, SMPT is used to transfer it from the client to the server and ultimately to the server of the recipient. However, the Message Access Agent facilitates the transmission of the message from the receiving server to the host server. POP3 and IMAP are the two types of protocols that are included in the Message Access Agent.





POP3 Ports

POP3 makes use of two network ports. They are:

- **Port 110:** Port 110 is a default TCP port used by POP3. But It has a disadvantage that it does not support encrypted communication.
- **Port 995:** Port 995 is majorly used for more secure applications. Port 995 is a [TLS](#) or [SSL](#) port used to provide more security.

28.3 Working of POP3

- Initially, POP3 needs to establish a connection between the POP client and the POP server.
- Once a secure connection is established several commands are exchanged between them to perform the task.
- Once a connection is established client requests available email messages.
- The Server sends the available messages along with their size and unique identifier number.
- Once the client receives the message, it makes a request to the server for downloading the messages. The user marks such messages and sends them to the server.
- Upon receiving from the client-server sends the messages selected by the client and accordingly marks them as read or unread.
- The client if want sends a request for deleting the messages.
- Once the tasks are completed the client sends a close connection request to the server

- The server then sends an acknowledgment to the client and closes the connection.

28.4 Advantages of POP3

- The message can be read offline also.
- POP3 requires less storage space.
- POP3 is easy to use and configure.
- POP3 is supported by many email applications.
- Since our PC already has the emails stored there, accessing them is quick and simple.
- The size of emails we send and receive is unrestricted.
- Since all emails are kept locally, less server storage space is needed.

28.5 Disadvantages of POP3

- POP3 does not provide the feature of real-time synchronization.
- If an email consists of virus attachments it can affect the system easily.
- POP3 does not support accessing the same email at the same time on different systems.
- At a time entire email folder can be discarded.
- It is also possible for the email folder that is downloaded from the mail server to get corrupted.
- Since the emails are kept locally, anyone using your computer can access the email folder.

28.6 Difference Between POP3 and IMAP

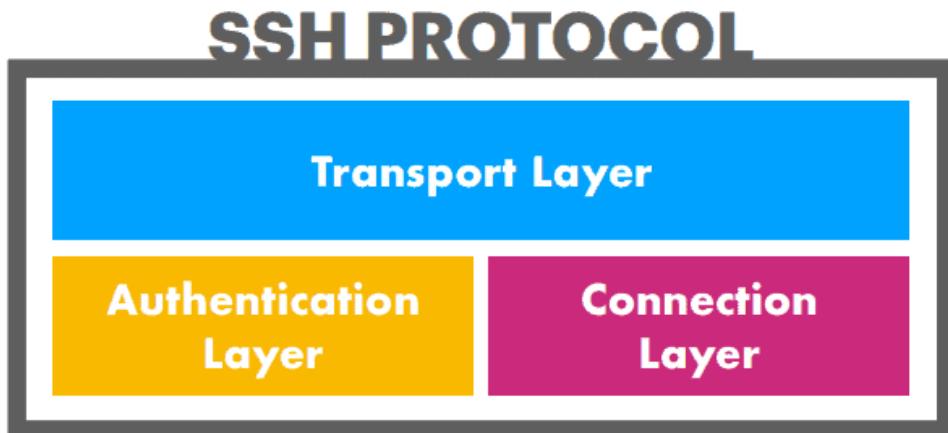
POP3	IMAP
POP is a simple protocol that only allows downloading messages from your Inbox to your local computer.	IMAP (Internet Message Access Protocol) is much more advanced and allows the user to see all the folders on the mail server.
The POP server listens on port 110, and the POP with SSL secure(POP3DS) server listens on port 995	The IMAP server listens on port 143, and the IMAP with SSL secure(IMAPDS) server listens on port 993.
In POP3 the mail can only be accessed from a single device at a time.	Messages can be accessed across multiple devices
To read the mail it has to be downloaded on the local system.	The mail content can be read partially before downloading
The user can not create, delete or rename email on the mail server.	The user can create, delete or rename an email on the mail server.

29. Layer 7 – SSH

[SSH](#) (short for Secure Shell) is a network protocol that provides a secure way for two computers to connect remotely. SSH employs [encryption](#) to ensure that hackers cannot interpret the traffic between two connected devices.

SSH consists of three distinct layers:

- **The transport layer** establishes safe and secure communication between a client and a server during and after [authentication](#). It oversees data encryption, decryption, and integrity protection. Furthermore, it helps speed up data exchange by providing data compression and caching.
- **The authentication layer** communicates the supported authentication methods to the client. It also conducts the entire user authentication process.
- **The connection layer** manages the communication between the machines after the authentication succeeds. It handles the opening and closing of communication channels and allows multiple channels for multiple sessions.



29.1 What is SSH Used for?

SSH provides a layer of security for information transfer between machines. Some important use cases for SSH are:

- [Remote access](#) – SSH ensures encrypted remote connections for users and processes.
- [File transfers](#) – SFTP, a secure file transfer protocol managed by SSH, provides a safe way to manipulate files over a network.
- [X11 Forwarding](#) – Users can run server-hosted X applications from their client machines.
- [Tunneling](#) – This [encapsulation technique](#) provides secure data transfers. [Tunneling](#) is useful for accessing business-sensitive online materials from unsecured networks, as it can act as a handy [VPN](#) alternative.
- [Network management](#) – The SSH protocol manages network infrastructure and other parts of the system.

- [Port Forwarding](#) – By mapping a client's port to the server's remote ports, SSH helps secure other network protocols, such as TCP/IP.

SSH is a client-server based protocol. This means the protocol allows a device requesting information or services (the client) to connect to another device (the server).

When a client connects to a server over SSH, the machine can be controlled like a local computer.

The server has a designated TCP port over which it monitors the network, waiting for clients to initialize the connection. Before a client connects and starts [issuing SSH commands](#), it needs to pass the authentication process.

29.2 Establishing an SSH Connection

Run the following command on a client machine to initiate an SSH connection:

```
ssh [username]@[server_ip_or_hostname]
```

When the server receives the requests, a session encryption negotiation begins.

Session Encryption Negotiation

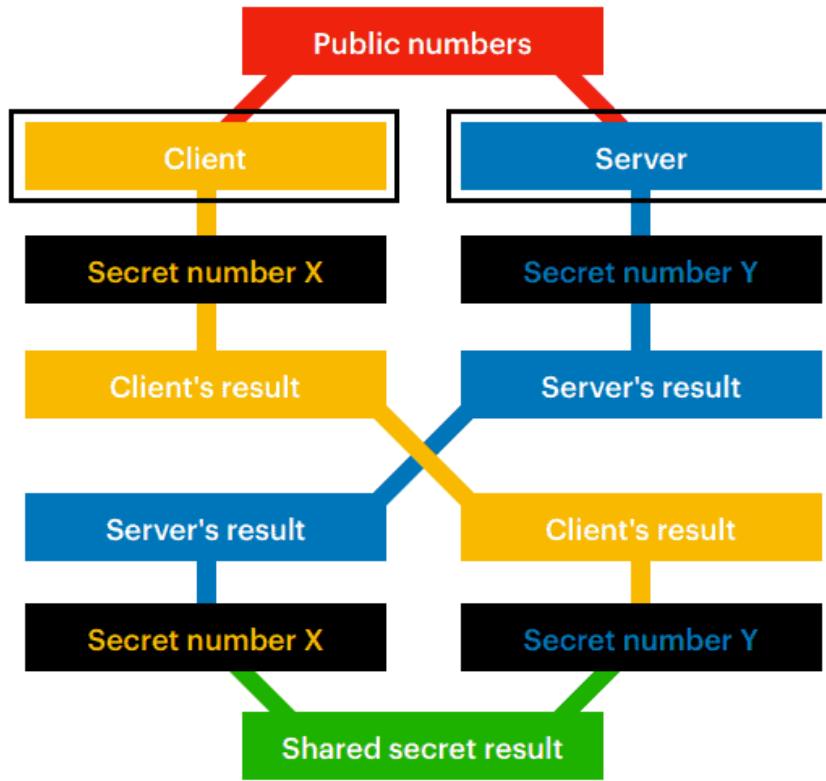
Upon receiving a connection request, the server sends the client a set of supported encryption protocols. The server uses the public key as the authentication method.

The client compares the protocols to its own set. If there are matching protocols, the machines agree to use one to establish the connection.

The client compares the server's public key to the stored private key stored in its system on the first connection attempt. If the keys match, the client and the server agree to use symmetric encryption to communicate during the SSH session. For this purpose, they communicate using an asymmetrically encrypted process that employs the [Diffie-Hellman \(DH\) key exchange algorithm](#).

The DH algorithm enables machines to work together and securely create a cryptographic key over a public network. To generate a key, the machines perform the following steps:

- **The machines agree on two numbers:** a modulus and a base number. To prevent brute force key decryption, the chosen modulus is a prime number of at least 600 digits.
- **The machines separately choose one number** and apply it to the equation involving the two public numbers.
- The server and the client **exchange the calculated values**.
- Each machine now **performs a calculation** using the result received from the other.



By performing the steps above, both machines calculate the same value, their [secret key](#). Finally, the server then attempts to authenticate the user who requests access.

Note: Strong user authentication is one measure to apply to secure your SSH server. For more security tips, read our [Linux SSH Security Best Practices](#) article.

User Authentication

The two most common SSH user authentication methods used are passwords and SSH keys. The clients safely send encrypted passwords to the server. However, passwords are a risky authentication method because their strength depends on the user's awareness of what makes a strong password.

Asymmetrically encrypted SSH public-private key pairs are a better option. Once the client decrypts the message, the server grants the client access to the system.

To generate an SSH key pair, type `ssh-keygen` in the terminal. As a result, the system generates and stores the keys.

The Internet Message Access Protocol (IMAP) serves as a cornerstone of cutting-edge email communication, facilitating seamless get admission to email messages. As a necessary element of the e-mail infrastructure, IMAP revolutionizes the manner

customers interact with their digital correspondence. Unlike its predecessor, the Post Office Protocol (POP), IMAP gives a dynamic and synchronized approach to handling emails across multiple gadgets and structures.

29.3 What is IMAP?

Internet Message Access Protocol (IMAP) is an application layer protocol that operates as a contract for receiving emails from the mail server. It was designed by Mark Crispin in 1986 as a remote access mailbox protocol, the current version of [IMAP](#) is IMAP4. It is used as the most commonly used protocol for retrieving emails. This term is also known as Internet mail access protocol, Interactive mail access protocol, and Interim mail access protocol. IMAP retrieves all of your recent messages from your email provider by getting in touch with it. They are removed from the email service as soon as you download them to your Mac or PC. This implies that the email can only be viewed on the same computer after it has been downloaded. You won't be able to access the previously downloaded messages if you attempt to access your email on a different device.

Features of IMAP

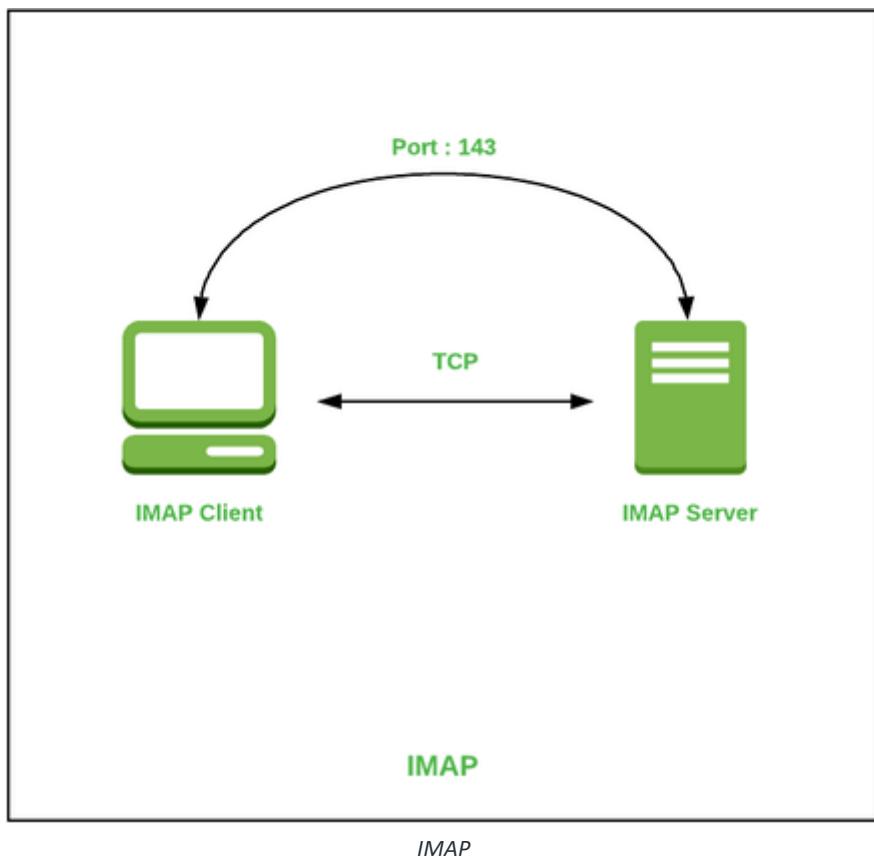
- It is capable of managing multiple mailboxes and organizing them into various categories.
- Provides adding of message flags to keep track of which messages are being seen.
- It is capable of deciding whether to retrieve email from a mail server before downloading.
- It makes it easy to download media when multiple files are attached.

29.4 Working of IMAP

IMAP follows Client-server Architecture and is the most commonly used email protocol. It is a combination of client and server process running on other computers that are connected through a network. This protocol resides over the TCP/IP protocol for communication. Once the communication is set up the server listens on port 143 by default which is non-encrypted. For the secure encrypted communication port, 993 is used.

29.5 Architecture of IMAP

The Internet Message Access Protocol (IMAP) protocol is a client-server model that allows users to access and view email messages stored on remote servers. Here is a summary of the events:



- **IMAP clients:** An IMAP client is an email application or software that users use to communicate with their email accounts. Examples include Microsoft Outlook, Mozilla Thunderbird, Apple Mail, and mobile email applications. The client communicates with the IMAP server to receive, manage, and send email messages.
- **IMAP Server:** The IMAP server manages email messages and manages user mailboxes. It responds to requests from IMAP clients, and provides access to email folders and messages. The server stores emails in a structured format, usually organized in user-defined folders or mailboxes. Common IMAP server software includes Dovecot, Courier IMAP, Cyrus IMAP, and Microsoft Exchange Server.
- **Network Protocols:** IMAP works over [TCP/IP](#) (Transmission Control Protocol/Internet Protocol) networks, and allows an IMAP client to connect to an IMAP server over the Internet or local area networks. IMAP typically uses TCP port 143 for unencrypted connections and TCP port 993 for encrypted connections using [SSL/TLS](#) (IMAPS).

29.6 Steps involve in IMAP Operation

- An email client, like Microsoft Outlook, connects to the server via IMAP when a user registers in.
- Certain ports are used for connections.

- The email client shows the headers of every email.
- IMAP does not automatically download attachments; messages are downloaded to the client only when the user taps on them.
- Compared to alternative email retrieval protocols like Post Office Protocol 3, users can check their mail more quickly with IMAP ([POP3](#)).
- Until they are specifically deleted by the user, emails will stay on the server.
- While IMAP over Secure Sockets Layer (SSL)/Transport Layer Security assigns port number 993, the IMAP server listens on port 143.

30. Layer 7 – NTP

NTP uses the UDP transport protocol over port 123. It uses UDP because NTP doesn't require a TCP connection, and using UDP results in low network overhead and low service response latency, so it is faster than TCP.

Network Time Protocol (NTP) is a protocol that helps the computers clock times to be synchronized in a network. This protocol is an application protocol that is responsible for the synchronization of hosts on a TCP/IP network. NTP was developed by David Mills in 1981 at the University of Delaware. This is required in a communication mechanism so that a seamless connection is present between the computers.

Features of NTP :

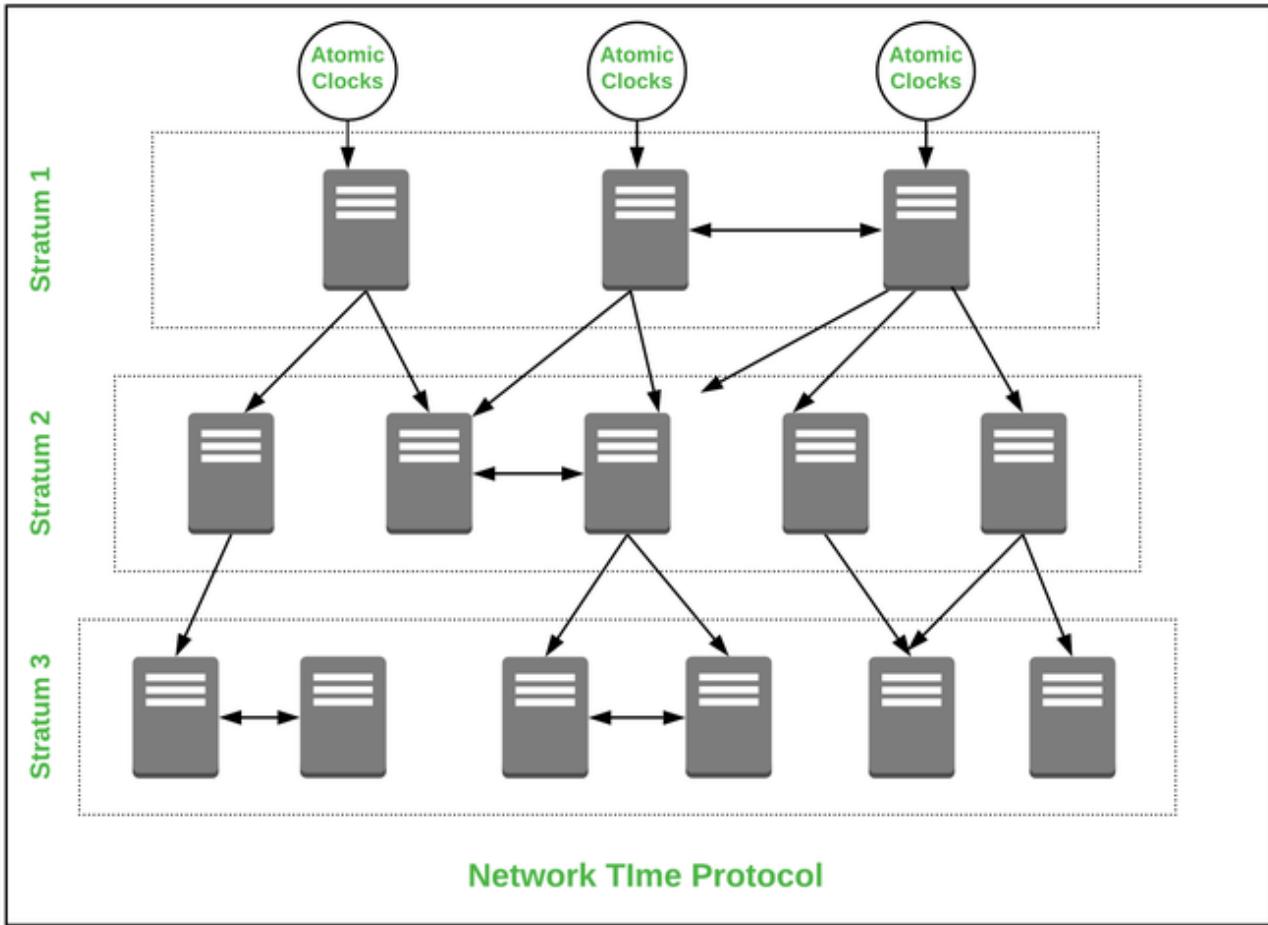
Some features of NTP are –

- NTP servers have access to highly precise atomic clocks and GPU clocks
- It uses Coordinated Universal Time (UTC) to synchronize CPU clock time.
- Avoids even having a fraction of vulnerabilities in information exchange communication.
- Provides consistent timekeeping for file servers

Working of NTP :

NTP is a protocol that works over the application layer, it uses a hierarchical system of time resources and provides synchronization within the stratum servers. First, at the topmost level, there is highly accurate time resources' ex. atomic or GPS clocks. These clock resources are called stratum 0 servers, and they are linked to the below NTP server called Stratum 1,2 or 3 and so on. These servers then provide the accurate date and time so that communicating hosts are synced to each other.

Architecture of Network Time Protocol :



Applications of NTP :

- Used in a production system where the live sound is recorded.
- Used in the development of Broadcasting infrastructures.
- Used where file system updates needed to be carried out across multiple computers depending on synchronized clock times.
- Used to implement security mechanism which depend on consistent time keeping over the network.
- Used in network acceleration systems which rely on timestamp accuracy to calculate performance.

Advantages of NTP :

- It provides internet synchronization between the devices.
- It provides enhanced security within the premises.
- It is used in the authentication systems like Kerberos.
- It provides network acceleration which helps in troubleshooting problems.
- Used in file systems that are difficult in network synchronization.

31. Layer 7 – Telnet

TELNET stands for **Teletype Network**. It is a type of protocol that enables one computer to connect to the local computer. It is used as a standard [TCP/IP protocol](#) for virtual terminal service which is provided by [ISO](#). The computer which starts the connection is known as the **local computer**.

The computer which is being connected to i.e. which accepts the connection known as the **remote computer**.

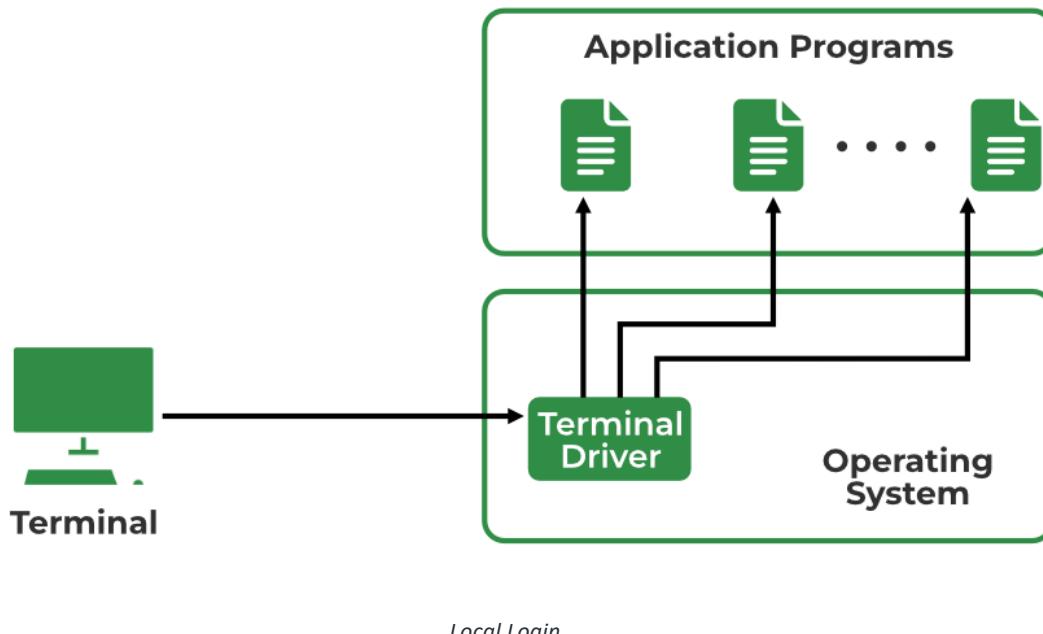
During telnet operation, whatever is being performed on the remote computer will be displayed by the local computer. Telnet operates on a client/server principle. The local computer uses a telnet client program and the remote computers use a telnet server program.

31.1 Logging

The logging process can be further categorized into two parts:

1. Local Login
2. Remote Login

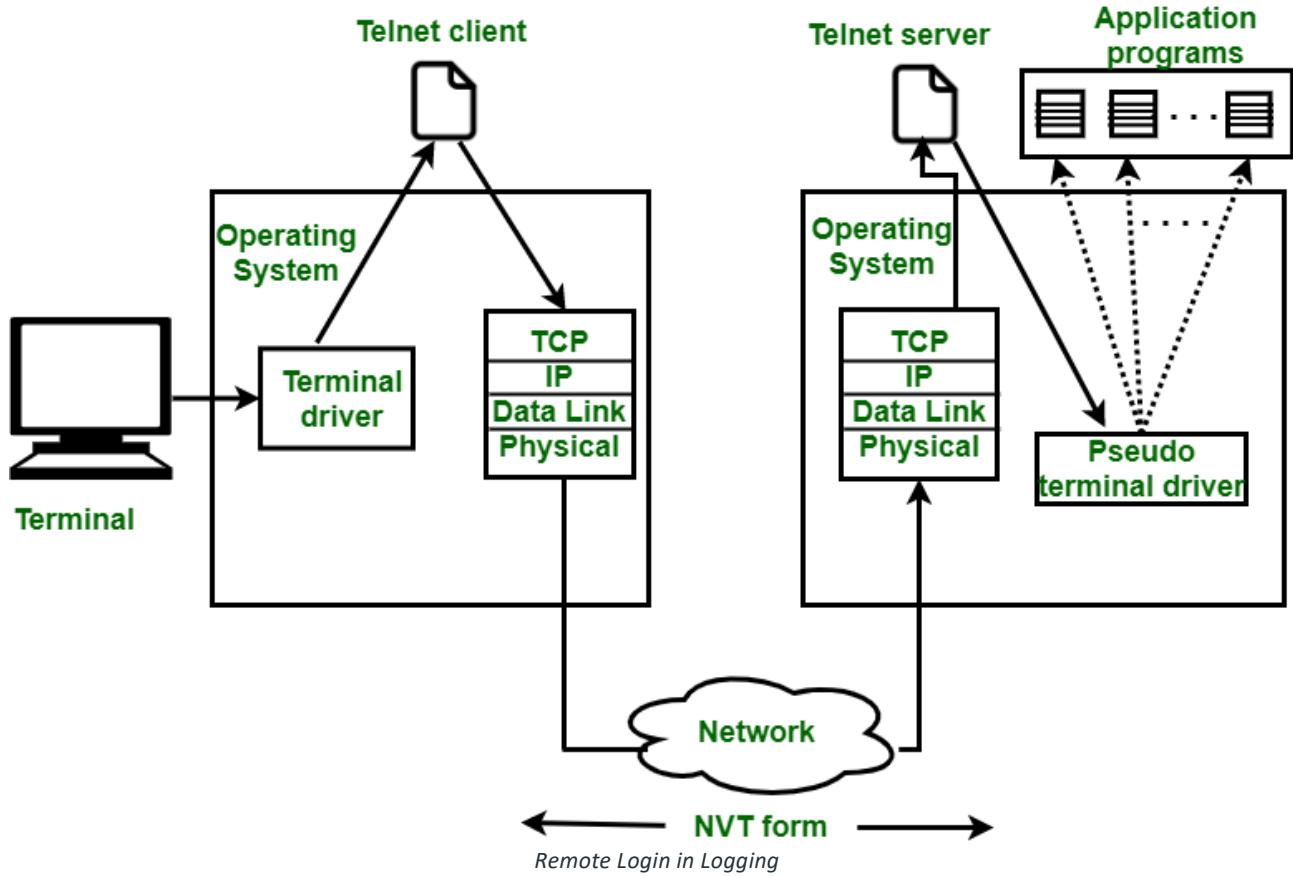
1. Local Login: Whenever a user logs into its local system, it is known as local login.



The Procedure of Local Login

- Keystrokes are accepted by the terminal driver when the user types at the terminal.
- Terminal Driver passes these characters to OS.
- Now, OS validates the combination of characters and opens the required application.

2. Remote Login: [Remote Login](#) is a process in which users can log in to a remote site i.e. computer and use services that are available on the remote computer. With the help of remote login, a user is able to understand the result of transferring the result of processing from the remote computer to the local computer.



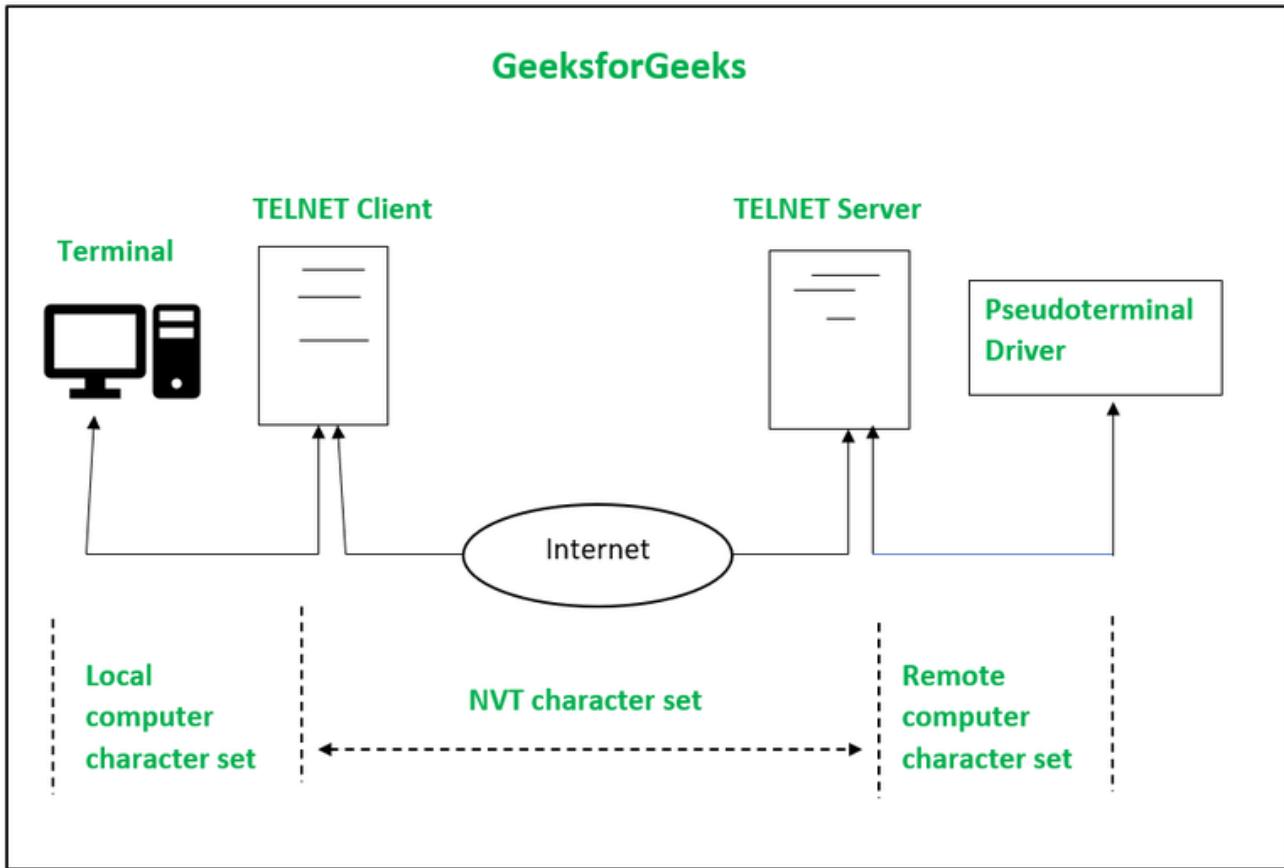
The Procedure of Remote Login

- When the user types something on the local computer, the local operating system accepts the character.
- The local computer does not interpret the characters, it will send them to the TELNET client.
- TELNET client transforms these characters to a universal character set called [Network Virtual Terminal \(NVT\)](#) characters and it will pass them to the local TCP/IP protocol Stack.
- Commands or text which are in the form of NVT, travel through the Internet and it will arrive at the [TCP/IP](#) stack at the remote computer.
- Characters are then delivered to the operating system and later on passed to the TELNET server.
- Then TELNET server changes those characters to characters that can be understandable by a remote computer.

- The remote operating system receives characters from a pseudo-terminal driver, which is a piece of software that pretends that characters are coming from a terminal.
- The operating system then passes the character to the appropriate application program.

31.2 Network Virtual Terminal(NVT)

NVT (Network Virtual Terminal) is a virtual terminal in TELNET that has a fundamental structure that is shared by many different types of real terminals. NVT (Network Virtual Terminal) was created to make communication viable between different types of terminals with different operating systems.



Network Virtual Terminal(NVT) in Telnet

31.3 TELNET Commands

Commands of Telnet are identified by a prefix character, Interpret As Command (IAC) with code 255. IAC is followed by command and option codes.

The basic format of the command is as shown in the following figure :



Character	Decimal	Binary	Meaning
WILL	251	11111011	1. Offering to enable. 2. Accepting a request to enable.
WON'T	252	11111100	1. Rejecting a request to enable. 2. Offering to disable. 3. Accepting a request to disable.
DO	253	11111101`	1. Approving a request to enable. 2. Requesting to enable.
DON'T	254	11111110	1. Disapproving a request to enable. 2. Approving an offer to disable. 3. Requesting to disable.

Following are some common options used with the telnet:

Code	Option	Meaning
0	Binary	It interprets as 8-bit binary transmission.
1	Echo	It will echo the data that is received on one side to the other side.
3	Suppress go ahead	It will suppress go ahead signal after data.
5	Status	It will request the status of TELNET.
6	Timing mark	It defines the timing marks.
8	Line width	It specifies the line width.

Code	Option	Meaning
9	Page size	It specifies the number of lines on a page.
24	Terminal type	It set the terminal type.
32	Terminal speed	It set the terminal speed.
34	Line mode	It will change to the line mode.

32. Layer 7 – FTP File Transfer Protocol (FTP) in Application Layer

File Transfer Protocol(FTP) is an application layer protocol that moves files between local and remote file systems. It runs on top of TCP, like HTTP. To transfer a file, 2 TCP connections are used by FTP in parallel: control connection and data connection.

32.1 What is File Transfer Protocol?

FTP is a standard communication protocol. There are various other protocols like HTTP which are used to transfer files between computers, but they lack clarity and focus as compared to FTP. Moreover, the systems involved in connection are heterogeneous, i.e. they differ in operating systems, directories, structures, character sets, etc the FTP shields the user from these differences and transfers data efficiently and reliably. FTP can transfer ASCII, EBCDIC, or image files. The ASCII is the default file share format, in this, each character is encoded by NVT ASCII. In ASCII or EBCDIC the destination must be ready to accept files in this mode. The image file format is the default format for transforming binary files.

FTP



Use Case :- *Upload / Download Files*

File Transfer Protocol



32.2 Types of FTP

There are different ways through which a server and a client do a file transfer using FTP. Some of them are mentioned below:

- **Anonymous FTP:** Anonymous FTP is enabled on some sites whose files are available for public access. A user can access these files without having any username or password. Instead, the username is set to anonymous, and the password is to the guest by default. Here, user access is very limited. For example, the user can be allowed to copy the files but not to navigate through directories.

- **Password Protected FTP:** This type of FTP is similar to the previous one, but the change in it is the use of username and password.
- **FTP Secure (FTPS):** It is also called as FTP Secure Sockets Layer (FTP SSL). It is a more secure version of FTP data transfer. Whenever FTP connection is established, Transport Layer Security (TLS) is enabled.
- **FTP over Explicit SSL/TLS (FTPES):** FTPES helps by upgrading FTP Connection from port 21 to an encrypted connection.
- **Secure FTP (SFTP):** SFTP is not a FTP Protocol, but it is a subset of Secure Shell Protocol, as it works on port 22.

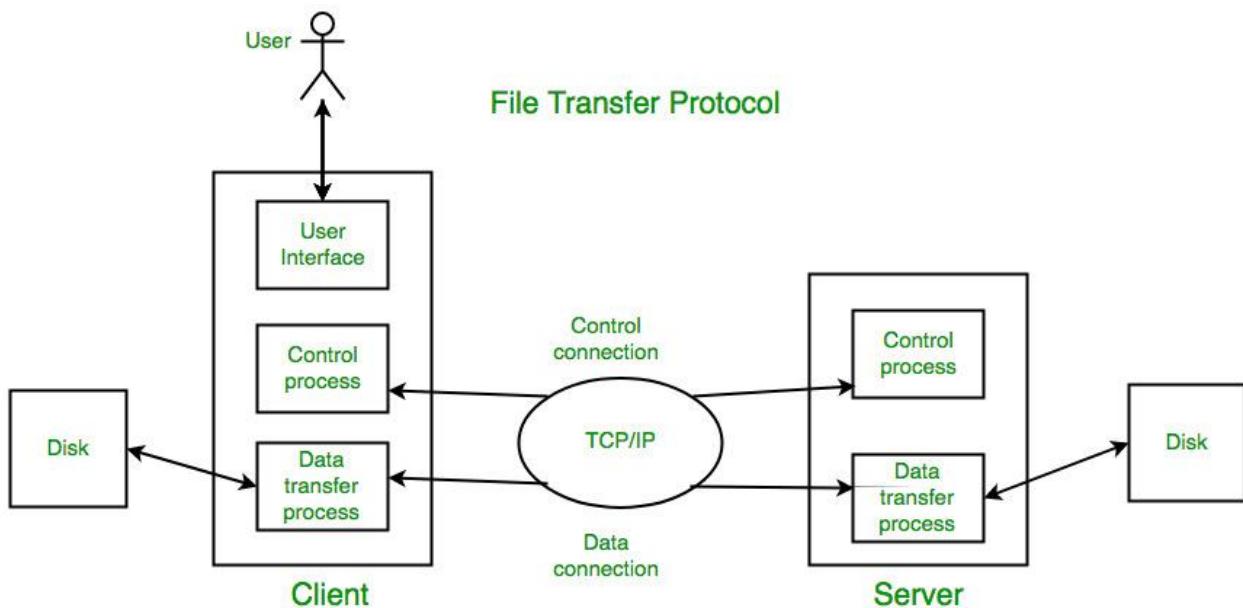
32.3 How Does FTP Work?

FTP is a client server protocol that has two communication channel, command channel for conversation control and data channel for file content.

Here are steps mentioned in which FTP works:

- A user has to log in to FTP Server first, there may be some servers where you can access to content without login, known as anonymous FTP.
- Client can start a conversation with server, upon requesting to download a file.
- The user can start different functions like upload, delete, rename, copy files, etc. on server.

FTP can work on different modes like Active and Passive modes



32.4 Types of Connection in FTP

- Control Connection
- Data Connection

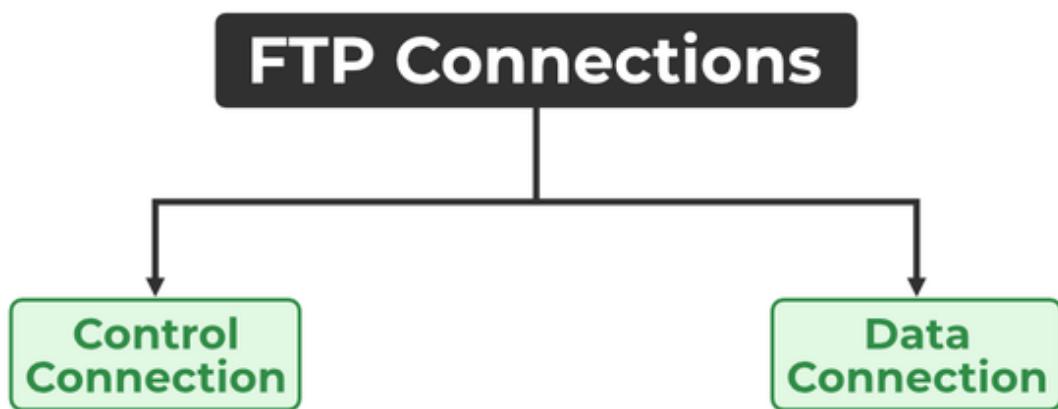
Control Connection

For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of a control connection. The control connection is initiated on port number 21.

Data connection

For sending the actual file, FTP makes use of a data connection. A data connection is initiated on port number 20.

FTP sends the control information out-of-band as it uses a separate control connection. Some protocols send their request and response header lines and the data in the same TCP connection. For this reason, they are said to send their control information in-band. HTTP and [SMTP](#) are such examples.



32.5 FTP Session

When an FTP session is started between a client and a server, the client initiates a control [TCP](#) connection with the server side. The client sends control information over this. When the server receives this, it initiates a data connection to the client side. But the control connection remains active throughout the user session. As we know HTTP is stateless . But FTP needs to maintain a state about its user throughout the session.

32.6 FTP Clients

FTP works on a [client-server model](#). The FTP client is a program that runs on the user's computer to enable the user to talk to and get files from remote computers. It is a set of commands that establishes the connection between two hosts, helps to transfer the files, and then closes the connection.

Some of the commands are:

get the filename(retrieve the file from the server)

get the filename(retrieve multiple files from the server)

ls(list files available in the current directory of the server)

There are also built-in FTP programs, which makes it easier to transfer files and it does not require remembering the commands.

32.7 FTP Data Types

The data type of a file, which determines how the file is represented overall, is the first piece of information that can be provided about it. The FTP standard specifies the following four categories of data:

- **ASCII:** Describes an ASCII text file in which each line is indicated by the previously mentioned type of end-of-line marker.
- **EBCDIC:** For files that use IBM's EBCDIC character set, this type is conceptually identical to ASCII.
- **Image:** This is the “black box” mode I described earlier; the file has no formal internal structure and is transferred one byte at a time without any processing.
- **Local:** Files containing data in logical bytes with a bit count other than eight can be handled by this data type.

32.8 FTP Replies

Some of the FTP replies are:

- 200 – Command okay.
- 530 – Not logged in.
- 331 – User name okay, need a password.
- 221 – Service closing control connection.
- 551 – Requested action aborted: page type unknown.
- 502 – Command not implemented.
- 503 – Bad sequence of commands.
- 504 – Command not implemented for that parameter.

32.9 Characteristics of FTP

- FTP uses TCP as a transport layer protocol.
- It is good for simple file transfers, such as during boot time.
- Errors in the transmission (lost packets, checksum errors) must be handled by the TFTP server.
- It uses only one connection through well-known port 69.
- [TFTP](#) uses a simple lock-step protocol (each data packet needs to be acknowledged). Thus the throughput is limited.

32.10 FTP's Security Issues

- Information could not go across a secure tunnel since FTP was not intended to do so. Thus, encryption is not present. A hacker would not need to struggle with encryption to access or alter data that is usable if they could intercept an FTP transaction.
- Even with FTP cloud storage, data can still be intercepted and misused if the service provider's system is attacked.
- As a result, data sent via FTP is a target for spoofing, sniffing, brute force, and other types of attacks that move somewhat slowly. A hacker might examine an FTP transmission and try to take advantage of any flaws by simply port scanning.

- The fact that FTP uses clear-text passwords—passwords that haven't been encrypted—is one of its main security flaws. Put differently, “Jerry1992” appears exactly like “Jerry1992.” The real password is hidden via an algorithm in more secure protocols. As a result, “Jerry1992” might appear as “dj18387saksng8937d9d8d7s6a8d89.” Passwords like this are not secured by FTP, which makes them more easily cracked by malicious actors.

32.11 Advantages of FTP

- File sharing also comes in the category of advantages of FTP in this between two machines files can be shared on the network.
- Speed is one of the main benefits of FTP.
- Since we don't have to finish every operation to obtain the entire file, it is more efficient.
- Using the username and password, we must log in to the FTP server. As a result, FTP might be considered more secure.
- We can move the files back and forth via FTP. Let's say you are the firm manager and you provide information to every employee, and they all reply on the same server.

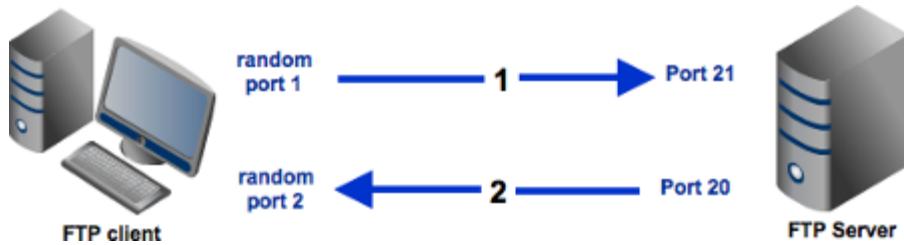
32.12 Disadvantages of FTP

- File size limit is the drawback of FTP only 2 GB size files can be transferred.
- More than one receivers are not supported by FTP.
- FTP does not encrypt the data this is one of the biggest drawbacks of FTP.
- FTP is unsecured we use login IDs and passwords making it secure but they can be attacked by hackers.

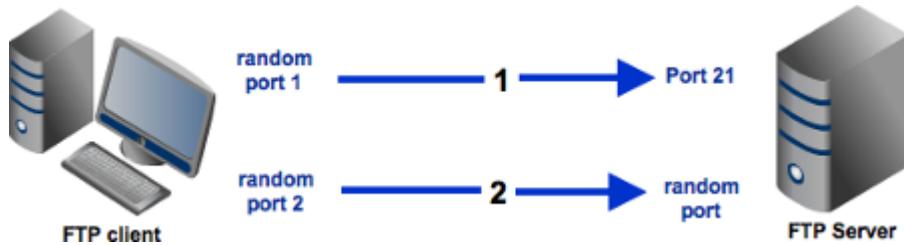
32.13 Difference Between FTP and SFTP

FTP	SFTP
It stands for File Transfer Protocol.	It stands for Secure File Transfer Protocol.
In FTP, secure channel is not provided to transfer the files between the hosts.	In SFTP, a secure channel is provided to transfer the files between the hosts.
It usually runs on port no-21.	It usually runs on port no-22.
It does not encrypt the data before sending	It encrypts data before sending.
It makes uploading and downloading of files without any security.	It maintains full security of the data by using SSH keys.

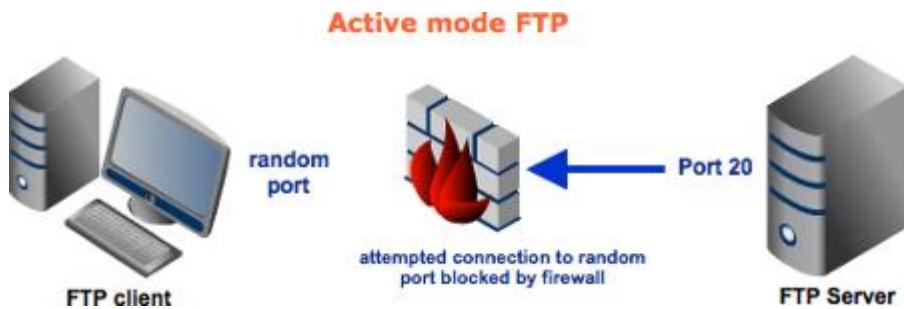
Active mode:



Passive mode:



In an active mode configuration, the server will attempt to connect to a random client-side port. So chances are, that port wouldn't be one of those predefined ports. As a result, an attempt to connect to it will be blocked by the firewall and no connection will be established.



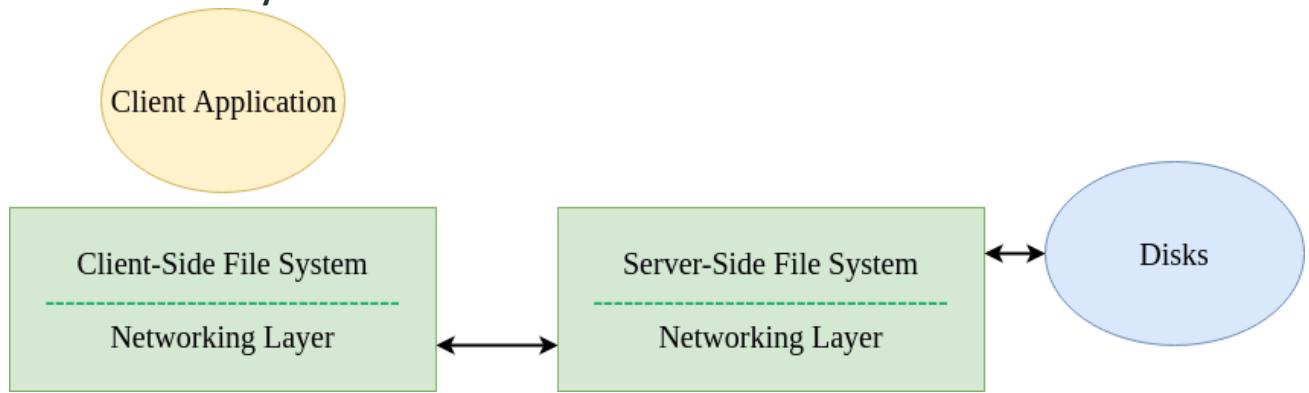
33. Layer 7 – NFS

Network File System (NFS – both tcp/udp) is used by UNIX clients for file access. NFS uses port 2049. NFSv3 and NFSv2 use the portmapper service on TCP or UDP port 111. T

The advent of distributed computing was marked by the introduction of distributed file systems. Such systems involved multiple client machines and one or a few servers. The server stores data on its disks and the clients may request data through some protocol messages. **Advantages of a distributed file system:**

- Allows easy sharing of data among clients.
- Provides centralized administration.
- Provides security, i.e. one must only secure the servers to secure data.

Distributed File System Architecture:



Even a simple client/server architecture involves more components than the physical file systems discussed previously in OS. The architecture consists of a **client-side file system** and a **server-side file system**. A client application issues a **system call** (e.g. `read()`, `write()`, `open()`, `close()` etc.) to access files on the client-side file system, which in turn retrieves files from the server. It is interesting to note that to a client application, the process seems no different than requesting data from a physical disk, since there is no special API required to do so. This phenomenon is known as **transparency** in terms of file access. It is the client-side file system that executes commands to service these system calls. For instance, assume that a client application issues the `read()` system call. The client-side file system then messages the server-side file system to read a block from the server's disk and return the data back to the client. Finally, it buffers this data into the `read()` buffer and completes the system call. The server-side file system is also simply called the **file server**.

Sun's Network File System: The earliest successful distributed system could be attributed to Sun Microsystems, which developed the Network File System (NFS). NFSv2 was the standard protocol followed for many years, designed with the goal of simple and fast server crash recovery. This goal is of utmost importance in multi-client and single-server based network architectures because a single instant of server crash means that all clients are unserviced. The entire system goes down. **Stateful protocols** make things complicated when it comes to crashes. Consider a client A trying to

access some data from the server. However, just after the first read, the server crashed. Now, when the server is up and running, client A issues the second read request. However, the server does not know which file the client is referring to, since all that information was temporary and lost during the crash. **Stateless protocols** come to our rescue. Such protocols are designed so as to not store any state information in the server. The server is unaware of what the clients are doing — what blocks they are caching, which files are opened by them and where their current file pointers are. The server simply delivers all the information that is required to service a client request. If a server crash happens, the client would simply have to retry the request. Because of their simplicity, NFS implements a stateless protocol. **File Handles:** NFS uses file handles to uniquely identify a file or a directory that the current operation is being performed upon. This consists of the following components:

- **Volume Identifier** – An NFS server may have multiple file systems or partitions. The volume identifier tells the server which file system is being referred to.
- **Inode Number** – This number identifies the file within the partition.
- Generation Number – This number is used while reusing an inode number.

File Attributes: “File attributes” is a term commonly used in NFS terminology. This is a collective term for the tracked metadata of a file, including file creation time, last modified, size, ownership permissions etc. This can be accessed by calling stat() on the file. **NFSv2 Protocol:** Some of the common protocol messages are listed below.

Message	Description
NFSPROC_GETATTR	Given a file handle, returns file attributes.
NFSPROC_SETATTR	Sets/updates file attributes.
NFSPROC_LOOKUP	Given file handle and name of the file to look up, returns file handle.
NFSPROC_READ	Given file handle, offset, count data and attributes, reads the data.
NFSPROC_WRITE	Given file handle, offset, count data and attributes, writes data into the file.
NFSPROC_CREATE	Given the directory handle, name of file and attributes, creates a file.
NFSPROC_REMOVE	Given the directory handle and name of file, deletes the file.

Message	Description
NFSPROC_MKDIR	Given directory handle, name of directory and attributes, creates a new directory.

The LOOKUP protocol message is used to obtain the file handle for further accessing data. The NFS mount protocol helps obtain the directory handle for the root (/) directory in the file system. If a client application opens a file /abc.txt, the client-side file system will send a LOOKUP request to the server, through the root (/) file handle looking for a file named abc.txt. If the lookup is successful, the file attributes are returned. **Client-Side Caching:** To improve performance of NFS, distributed file systems cache the data as well as the metadata read from the server onto the clients. This is known as client-side caching. This reduces the time taken for subsequent client accesses. The cache is also used as a temporary buffer for writing. This helps improve efficiency even more since all writes are written onto the server at once.

34. Glossary

Bit: The smallest representation of data that a computer can understand

Border Gateway Protocol (BGP): A protocol by which routers share data with each other

Broadcast: A type of Ethernet transmission, sent to every single device on a LAN

Broadcast address: A special destination used by an Ethernet broadcast composed by all Fs

Cable categories: Groups of cables that are made with the same material. Most network cables used today can be split into two categories, copper and fiber

Cables: Insulated wires that connect different devices to each other allowing data to be transmitted over them

Carrier-Sense Multiple Access with Collision Detection (CSMA/CD): CSMA/CD is used to determine when the communications channels are clear and when the device is free to transmit data

Client: A device that receives data from a server

Collision domain: A network segment where only one device can communicate at a time

Computer networking: The full scope of how computers communicate with each other

Copper cable categories : These categories have different physical characteristics like the number of twists in the pair of copper wires. These are defined as names like category (or cat) 5, 5e, or 6, and how quickly data can be sent across them and how resistant they are to outside interference are all related to the way the twisted pairs inside are arranged

Crosstalk: Crosstalk is when an electrical pulse on one wire is accidentally detected on another wire

Cyclical Redundancy Check (CRC): A mathematical transformation that uses polynomial division to create a number that represents a larger set of data. It is an important concept for data integrity and is used all over computing, not just network transmissions

Data packet: An all-encompassing term that represents any single set of binary data being sent across a network link

Datalink layer: The layer in which the first protocols are introduced. This layer is responsible for defining a common way of interpreting signals, so network devices can communicate

Destination MAC address: The hardware address of the intended recipient that immediately follows the start frame delimiter

Duplex communication: A form of communication where information can flow in both directions across a cable

Ethernet: The protocol most widely used to send data across individual links

Ethernet frame: A highly structured collection of information presented in a specific order

EtherType field: It follows the Source MAC Address in a dataframe. It's 16 bits long and used to describe the protocol of the contents of the frame

Fiber cable: Fiber optic cables contain individual optical fibers which are tiny tubes made of glass about the width of a human hair. Unlike copper, which uses electrical voltages, fiber cables use pulses of light to represent the ones and zeros of the underlying data

Five layer model: A model used to explain how network devices communicate. This model has five layers that stack on top of each other: Physical, Data Link, Network, Transport, and Application

Frame check sequence: It is a 4-byte or 32-bit number that represents a checksum value for the entire frame

Full duplex: The capacity of devices on either side of a networking link to communicate with each other at the exact same time

Half-duplex: It means that, while communication is possible in each direction, only one device can be communicating at a time

Hexadecimal: A way to represent numbers using a numerical base of 16

Hub: It is a physical layer device that broadcasts data to every computer connected to it

Internet Protocol (IP): The most common protocol used in the network layer

Internet Service Provider (ISP): A company that provides a consumer an internet connection

Internetwork: A collection of networks connected together through routers - the most famous of these being the Internet

Line coding: Modulation used for computer networks

Local Area Network (LAN): A single network in which multiple devices are connected

MAC(Media Access Control) address: A globally unique identifier attached to an individual network interface. It's a 48-bit number normally represented by six groupings of two hexadecimal numbers

Modulation: A way of varying the voltage of a constant electrical charge moving across a standard copper network cable

Multicast frame: If the least significant bit in the first octet of a destination address is set to one, it means you're dealing with a multicast frame. A multicast frame is similarly set to all devices on the local network signal, and it will be accepted or discarded by each device depending on criteria aside from their own hardware MAC address

Network layer: It's the layer that allows different networks to communicate with each other through devices known as routers. It is responsible for getting data delivered across a collection of networks

Network port: The physical connector to be able to connect a device to the network. This may be attached directly to a device on a computer network, or could also be located on a wall or on a patch panel

Network switch: It is a level 2 or data link device that can connect to many devices so they can communicate. It can inspect the contents of the Ethernet protocol data being sent around the network, determine which system the data is intended for and then only send that data to that one system

Node: Any device connected to a network. On most networks, each node will typically act as a server or a client

Octet: Any number that can be represented by 8 bits

Organizationally Unique Identifier (OUI): The first three octets of a MAC address

OSI model: A model used to define how network devices communicate. This model has seven layers that stack on top of each other: Physical, Data Link, Network, Transport, Session, Presentation, and Application

Patch panel: A device containing many physical network ports

Payload: The actual data being transported, which is everything that isn't a header

Physical layer: It represents the physical devices that interconnect computers

Preamble: The first part of an Ethernet frame, it is 8 bytes or 64 bits long and can itself be split into two sections

Protocol: A defined set of standards that computers must follow in order to communicate properly is called a protocol

Router: A device that knows how to forward data between independent networks

Server: A device that provides data to another device that is requesting that data, also known as a client

Simplex communication: A form of data communication that only goes in one direction across a cable

Source MAC address: The hardware address of the device that sent the ethernet frame or data packet. In the data packet it follows the destination MAC address

Start Frame Delimiter (SFD): The last byte in the preamble, that signals to a receiving device that the preamble is over and that the actual frame contents will now follow

Transmission Control Protocol (TCP): The data transfer protocol most commonly used in the fourth layer. This protocol requires an established connection between the client and server

Transport layer: The network layer that sorts out which client and server programs are supposed to get the data

Twisted pair cable: The most common type of cabling used for connecting computing devices. It features pairs of copper wires that are twisted together

Unicast transmission: A unicast transmission is always meant for just one receiving address

User Datagram Protocol (UDP): A transfer protocol that does not rely on connections. This protocol does not support the concept of an acknowledgement. With UDP, you just set a destination port and send the data packet

Virtual LAN (VLAN): It is a technique that lets you have multiple logical LANs operating on the same physical equipment

VLAN header: A piece of data that indicates what the frame itself is. In a data packet it is followed by the EtherType

Address class system: **A system which defines how the global IP address space is split up**

Address Resolution Protocol (ARP): **A protocol used to discover the hardware address of a node with a certain IP address**

ARP table: **A list of IP addresses and the MAC addresses associated with them**

ASN: **Autonomous System Number is a number assigned to an individual autonomous system**

Demarcate: **To set the boundaries of something**

Demarcation point: **Where one network or system ends and another one begins**

Destination network: **The column in a routing table that contains a row for each network that the router knows about**

DHCP: **A technology that assigns an IP address automatically to a new device. It is an application layer protocol that automates the configuration process of hosts on a network**

Dotted decimal notation: **A format of using dots to separate numbers in a string, such as in an IP address**

Dynamic IP address: An IP address assigned automatically to a new device through a technology known as Dynamic Host Configuration Protocol

Exterior gateway: Protocols that are used for the exchange of information between independent autonomous systems

Flag field: It is used to indicate if a datagram is allowed to be fragmented, or to indicate that the datagram has already been fragmented

Fragmentation: The process of taking a single IP datagram and splitting it up into several smaller datagrams

Fragmentation offset field: It contains values used by the receiving end to take all the parts of a fragmented packet and put them back together in the correct order

Header checksum field: A checksum of the contents of the entire IP datagram header

Header length field: A four bit field that declares how long the entire header is. It is almost always 20 bytes in length when dealing with IPv4

IANA: The Internet Assigned Numbers Authority, is a non-profit organization that helps manage things like IP address allocation

Identification field: It is a 16-bit number that's used to group messages together

Interface: For a router, the port where a router connects to a network. A router gives and receives data through its interfaces. These are also used as part of the routing table

Interior gateway: Interior gateway protocols are used by routers to share information within a single autonomous system

IP datagram: A highly structured series of fields that are strictly defined

IP options field: An optional field and is used to set special characteristics for datagrams primarily used for testing purposes

Network Address Translation (NAT): A mitigation tool that lets organizations use one public IP address and many private IP addresses within the network

Next hop: The IP address of the next router that should receive data intended for the destination networking question or this could just state the network is directly connected and that there aren't any additional hops needed. Defined as part of the routing table

Non-routable address space: They are ranges of IPs set aside for use by anyone that cannot be routed to

Padding field: A series of zeros used to ensure the header is the correct total size

Protocol field: A protocol field is an 8-bit field that contains data about what transport layer protocol is being used

Routing protocols: Special protocols the routers use to speak to each other in order to share what information they might have

Service type field: A eight bit field that can be used to specify details about quality of service or QoS technologies

Static IP address: An IP address that must be manually configured on a node

Subnet mask: 32-bit numbers that are normally written as four octets of decimal numbers

Subnetting: The process of taking a large network and splitting it up into many individual smaller sub networks or subnets

Time-To-Live field (TTL): An 8-bit field that indicates how many router hops a datagram can traverse before it's thrown away

Total hops: The total number of devices data passes through to get from its source to its destination. Routers try to choose the shortest path, so fewest hops possible. The routing table is used to keep track of this

Total length field: A 16-bit field that indicates the total length of the IP datagram it's attached to

ACK flag: One of the TCP control flags. ACK is short for acknowledge. A value of one in this field means that the acknowledgment number field should be examined

Acknowledgement number: The number of the next expected segment in a TCP sequence

Application layer: The layer that allows network applications to communicate in a way they understand

Application layer payload: The entire contents of whatever data applications want to send to each other

CLOSE: A connection state that indicates that the connection has been fully terminated, and that no further communication is possible

CLOSE_WAIT: A connection state that indicates that the connection has been closed at the TCP layer, but that the application that opened the socket hasn't released its hold on the socket yet

Connection-oriented protocol: A data-transmission protocol that establishes a connection at the transport layer, and uses this to ensure that all data has been properly transmitted

Connectionless protocol: A data-transmission protocol that allows data to be exchanged without an established connection at the transport layer. The most common of these is known as UDP, or User Datagram Protocol

Data offset field: The number of the next expected segment in a TCP packet/datagram

Demultiplexing: Taking traffic that's all aimed at the same node and delivering it to the proper receiving service

Destination port: The port of the service the TCP packet is intended for

ESTABLISHED: Status indicating that the TCP connection is in working order, and both sides are free to send each other data

FIN: One of the TCP control flags. FIN is short for finish. When this flag is set to one, it means the transmitting computer doesn't have any more data to send and the connection can be closed

FIN_WAIT: A TCP socket state indicating that a FIN has been sent, but the corresponding ACK from the other end hasn't been received yet

Firewall: It is a device that blocks or allows traffic based on established rules

FTP: An older method used for transferring files from one computer to another, but you still see it in use today

Handshake: A way for two devices to ensure that they're speaking the same protocol and will be able to understand each other

Instantiation: The actual implementation of something defined elsewhere

Listen: It means that a TCP socket is ready and listening for incoming connections

Multiplexing: It means that nodes on the network have the ability to direct traffic toward many different receiving services

Options field: It is sometimes used for more complicated flow control protocols

Port: It is a 16-bit number that's used to direct traffic to specific services running on a networked computer

Presentation layer: It is responsible for making sure that the unencapsulated application layer data is actually able to be understood by the application in question

PSH flag: One of the TCP control flags. PSH is short for push. This flag means that the transmitting device wants the receiving device to push currently-buffered data to the application on the receiving end as soon as possible

RST flag: One of the TCP control flags. RST is short for reset. This flag means that one of the sides in a TCP connection hasn't been able to properly recover from a series of missing or malformed segments

Sequence number: A 32-bit number that's used to keep track of where in a sequence of TCP segments this one is expected to be

Server or Service: A program running on a computer waiting to be asked for data

Session layer: The network layer responsible for facilitating the communication between actual applications and the transport layer

Socket: The instantiation of an endpoint in a potential TCP connection

Source port: A high numbered port chosen from a special section of ports known as ephemeral ports

SYN flag: One of the TCP flags. SYN stands for synchronize. This flag is used when first establishing a TCP connection and make sure the receiving end knows to examine the sequence number field

SYN RECEIVED: A TCP socket state that means that a socket previously in a listener state, has received a synchronization request and sent a SYN-ACK back

SYN_SENT: A TCP socket state that means that a synchronization request has been sent, but the connection hasn't been established yet

TCP checksum: A mechanism that makes sure that no data is lost or corrupted during a transfer

TCP segment: A payload section of an IP datagram made up of a TCP header and a data section

TCP window: The range of sequence numbers that might be sent before an acknowledgement is required

URG flag: One of the TCP control flags. URG is short for urgent. A value of one here indicates that the segment is considered urgent and that the urgent pointer field has more data about this

Urgent pointer field: A field used in conjunction with one of the TCP control flags to point out particular segments that might be more important than others

A record: The most common resource record, used to point a certain domain name at a certain IPv4 IP address

Anycast: A technique that's used to route traffic to different destinations depending on factors like location, congestion, or link health

Automatic allocation: A range of IP addresses is set aside for assignment purposes

Caching and recursive name servers: They are generally provided by an ISP or your local network, and their purpose is to store domain name lookups for a certain amount of time

CNAME: A resource record used to map one domain to another

DHCP discovery: The process by which a client configured to use DHCP attempts to get network configuration information

Domain Name System (DNS): A global and highly distributed network service that resolves strings of letters, such as a website name, into an IP address

DNS zones: A portion of space in the Domain Name System (DNS) that is controlled by an authoritative name server

Domain: Used to demarcate where control moves from a top-level domain name server to an authoritative name server

Domain name: A website name; the part of the URL following www.

Dynamic allocation: A range of IP addresses is set aside for client devices and one of these IPs is issued to these devices when they request one

Fixed allocation: Requires a manually specified list of MAC address and the corresponding IPs

Fully qualified domain name: When you combine all the parts of a domain together

IP masquerading: The NAT obscures the sender's IP address from the receiver

MX record: It stands for mail exchange and this resource record is used in order to deliver email to the correct server

Name resolution: This process of using DNS to turn a domain name into an IP address

Network Address Translation (NAT): A mitigation tool that lets organizations use one public IP address and many private IP addresses within the network

NS record: It indicates other name servers that may also be responsible for a particular zone

NTP servers: Used to keep all computers on a network synchronized in time

Pointer resource record: It resolves an IP to a name

Port forwarding: A technique where specific destination ports can be configured to always be delivered to specific nodes

Port preservation: A technique where the source port chosen by a client, is the same port used by the router

Proxy service: A server that acts on behalf of a client in order to access another service

Quad A (AAAA) record: It is very similar to an A record except that it returns in IPv6 address instead of an IPv4 address

Recursive name servers: Servers that perform full DNS resolution requests

Reverse lookup zone files: They let DNS resolvers ask for an IP, and get the FQDN associated with it returned

Reverse proxy: A service that might appear to be a single server to external clients, but actually represents many servers living behind it

Round robin: It is a concept that involves iterating over a list of items one by one in an orderly fashion

SRV record: A service record used to define the location of various specific services

Start of authority: A declaration of the zone and the name of the name server that is authoritative for it

Top Level Domain (TLD): The top level of the DNS or the last part of a domain name. For example, the "com" in www.weather.com

Time-To-Live field (TTL): An 8-bit field that indicates how many router hops a datagram can traverse before it's thrown away

Two-factor authentication: A technique where more than just a username and password are required to authenticate. Usually, a short-lived numerical token is generated by the user through a specialized piece of hardware or software

TXT record: It stands for text and was originally intended to be used only for associating some descriptive text with a domain name for human consumption

Types of DNS servers: There are five primary types of DNS servers; caching name servers, recursive name servers, root name servers, TLD name servers, and authoritative name servers

Virtual Private Network (VPN): A technology that allows for the extension of a private or local network, to a host that might not work on that same local network

Zone Files: Simple configuration files that declare all resource records for a particular zone

Ad-Hoc network: A network configuration without supporting network infrastructure. Every device involved with the ad-hoc network communicates with every other device within range, and all nodes help pass along messages

Asymmetric Digital Subscriber Line (ADSL): A device that establishes data connections across phone lines and different speeds for uploading and downloading data

Baud rate: A measurement of how many bits could be passed across a phone line in a second

Bluetooth: The most common short range wireless network

Broadband: Any connectivity technology that isn't dial-up Internet

Cable modem: A device that sits at the edge of a consumer's network and connects it to the cable modem termination system

Cable modem termination system: Connects lots of different cable connections to an ISP's core network

Channels: Individual, smaller sections of the overall frequency band used by a wireless network

Collision domain: A network segment where only one device can communicate at a time

Data payload section: Has all of the data of the protocols further up the stack of a frame

Dial-up: Uses POTS for data transfer, and gets its name because the connection is established by actually dialing a phone number

DSL: Digital subscriber line was able to send much more data across the wire than traditional dial-up technologies by operating at a frequency range that didn't interfere with normal phone calls

DSLAM: Digital Subscriber Line Access Multiplexers are devices that connect multiple DSL connections to a high-speed digital communications channel

Duration field: Specifies how long the total frame is

Frame check sequence: It is a 4-byte or 32-bit number that represents a checksum value for the entire frame

Frame control field: 16 bits long, it contains a number of sub-fields that are used to describe how the frame itself should be processed

Frequency band: A certain section of the radio spectrum that's been agreed upon to be used for certain communications

FTTB: Fiber to the building, fiber to the business or even fiber to the basement, since this is generally where cables to buildings physically enter. FTTB is a setup where fiber technologies are used for data delivery to an individual building

FTTH: Fiber to the home. This is used in instances where fiber is actually run to each individual residents in a neighborhood or apartment building

FTTN: Fiber to the neighborhood. This means that fiber technologies are used to deliver data to a single physical cabinet that serves a certain amount of the population

FTTP: Fiber to the premises. FTTH and FTTB may both also be referred to as FTTP

FTTX: Stands for fiber to the X, where the X can be one of many things

HDSL: High Bit-rate Digital Subscriber Lines. These are DSL technologies that provision speeds above 1.544 megabits per second

MAC filtering: Access points are configured to only allow for connections from a specific set of MAC addresses belonging to devices you trust

Mesh networks: Like ad-hoc networks, lots of devices communicate with each other device, forming a mesh if you were to draw lines for all the links between all the nodes

Metered connection: An internet connection where all data transfer usage is tracked. Cell phone plans that have a limit on data usage per month or that charge based on usage are examples of metered connections

Non-metered connection: A connection where your data usage is not tracked or limited, instead you are charged a flat fee for unlimited and unrestricted usage. A Wi-Fi connection is an example of a non-metered connection

Optical Network Terminator: Converts data from protocols the fiber network can

understand to those that are more traditional twisted pair copper networks can understand

Pairing: When a wireless peripheral connects to a mobile device, and the two devices exchange information, sometimes including a PIN or password, so that they can remember each other

Point-To-Point VPN: Establishes a VPN tunnel between two sites but VPN tunneling logic is handled by network devices at either side, so that users don't all have to establish their own connections

Receiving address: The MAC address of the access point that should receive the frame

Symmetric Digital Subscriber Line (SDSL): A device that establishes data connections across phone lines and has upload and download speeds that are the same

Sequence control field: A field that is 16 bits long and mainly contains a sequence number used to keep track of ordering the frames

Short-range wireless network: It is what mobile devices uses to connect to their peripherals

T-Carrier technologies: Technologies Invented to transmit multiple phone calls over a single link. Eventually, they also became common transmission systems to transfer data much faster than any dial-up connection could handle

Transmitter address: The MAC address of whatever has just transmitted the frame

Wi-Fi Protected Access (WPA): A security program that uses a 128-bit key to protect wireless computer networks, which makes it more difficult to crack than WEP

Wide area network: Acts like a single network but spans across multiple physical locations. WAN technologies usually require that you contract a link across the Internet with your ISP

Wired Equivalence Privacy (WEP): An encryption technology that provides a very low level of privacy. WEP should really only be seen as being as safe as sending unencrypted data over a wired connection

Wireless access point: A device that bridges the wireless and wired portions of a network

Wireless LANS (WLANS): One or more access points act as a bridge between a wireless and a wired network

Wireless networking: Networks you connect to through radios and antennas

Cloud computing: The concept and technological approach of accessing data, using applications, storing files, etc. from anywhere in the world as long as you have an internet connection

Error detection: The ability for a protocol or program to determine that something went wrong

Error recovery: The ability for a protocol or program to attempt to fix an error

Flow label field: 20-bit field that's used in conjunction with the traffic class field for routers to make decisions about the quality of service level for a specific datagram

Hop limit field: An 8-bit field that's identical in purpose to the TTL field in an IPv4 header

Host file: A simple text file that consists of one or more lines, each of which contains a hostname and an IP address

Hybrid cloud: A cloud computing environment that combines on-premises private cloud resources with public cloud resources

Hypervisor: A piece of software that runs and manages virtual machines while also offering guests a virtual operating platform that's indistinguishable from actual hardware

ICMP: Internet control message protocol is used by router or remote hosts to communicate error messages when network problems prevent delivery of IP packets

ICMP payload: Piece of the packet which lets the recipient of the message know which of their transmissions caused the error being reported

Infrastructure as a Service (IaaS): A subset of cloud computing where a network and servers are provided for customers to run their services

IPv6 tunnel brokers: Companies that provide IPv6 tunneling endpoints for you, so you don't have to introduce additional equipment to your network

IPv6 tunnel: IPv6 tunnel servers on either end of a connection take incoming IPv6 traffic and encapsulate it within traditional IPv4 datagrams

Link-local unicast address: Allow for local network segment communications and are configured based upon a host's MAC address

Loopback address: An IP address that always points to itself. This type of address is used to test internal pathing through the TCP/IP protocols

Multicast: A way of addressing groups of hosts all at once

Next header field: Defines what kind of header is immediately after this current one

Payload length field: 16-bit field that defines how long the data payload section of the datagram is

Platform as a service: A subset of cloud computing where a platform is provided for customers to run their services

Private cloud: When a company owns the services and the rest of the cloud infrastructure, whether on-site or in a remote data center

Public cloud: The cloud services provided by a third party

Public DNS servers: Name servers specifically set up so that anyone can use them for free

Registrar: An organization responsible for assigning individual domain names to other organizations or individuals

Software as a Service (SaaS): A way of licensing the use of software to others while keeping that software centrally hosted and managed

Traffic class field: An 8-bit field that defines the type of traffic contained within the IP datagram and allows for different classes of traffic to receive different priorities

Version field: First field in an IP header that specifies the version of IP

Virtualization: A single physical machine called a host runs many individual virtual instances called guests

<https://spun.io/2018/05/04/bitwise-operations-and-common-uses-decoding-ipv4-headers-using-and-and-bitwise-shifts/>