Networking

Created	@January 24, 2022 2:49 PM
Reviewed	☐

Table of contents

Table of contents
Local Area Network (LAN)
Metropolitan area network
OSI model
TCP/IP model
Encapsulation
Transport layer
Network layer
Subnetting
Data-Link Layer
Physical layer
Network devices
Life of a packet
Network troubleshooting
SNMP
SSH
Basics & tcpdump

NAT
Ports
VOIP

Networking

What is network?

A computer network can be described as a system of interconnected devices that can communicate using some common standards (called **protocols**).

▼ Local Area Network (LAN)

The term **local area network (LAN)** is commonly used to describe a network of devices in a limited area (a house, office, building…). This type of network is usually capable of achieving high data transfer rate (up to 10 Gbps!) at low cost. Examples of this type of network are a small office network inside a single building or your home network.

A typical SOHO (small office/home office) LAN consist of **PCs, printers, switches, routers**, and cabling that connects all these devices together.

2 computers connected to **switch** which is connected to the **router** that provides the LAN with access to the Internet.

Some of the most popular LAN technologies are Ethernet, Token Ring and FDDI. Most LAN networks use TCP/IP to communicate.

WLANs use radio waves instead of wires or cables for links between devices.

▼ Metropolitan area network

The term **metropolitan area network** is used to describe a network in a single metropolitan area, hence the name. This type of network is usually bigger than a LAN and smaller than a WAN. An example of this type of network would be a network that connects two company offices inside the same city.

▼ OSI model

OSI (**Open Systems Interconnection**) model was created by the **International Organization for Standardization (ISO)**, an international standard-setting body. It was designed to be a reference model for describing the functions of a communication system.

Application	Away
Presentation	Pizza
Session	Sausage
Transport	Throw
Network	Not
Data Link	Do
Physical	Please

7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

Physical – defines how to move bits from one device to another. It details how **cables, connectors and network interface cards** are supposed to work and how to send and receive bits.

Data Link – encapsulates a packet in a frame. A frame contains a header and a trailer that enable devices to communicate. A header (most commonly) contains a source and destination **MAC address**. A trailer contains the Frame Check Sequence field, which is used to **detect transmission errors** (only detection no correction). The data link layer has two sublayers:

1. **Logical Link Control** – used for flow control and error detection.
2. **Media Access Control** – used for hardware addressing and for controlling the access method.

Network – defines device addressing, routing, and path determination. Device (logical) addressing is used to identify a host on a network (e.g. by its **IP address**).

Transport – segments big chunks of data received from the upper layer protocols. Establishes and terminates connections between two computers. Used for flow control

and data recovery. Works on **TCP and UDP** protocol. It also carries the **error correction.**

Session – defines how to **establish and terminate a session** between the two systems.

Presentation – Data conversation from standard to machine language. Encryption, decryption, and compression also takes place over here. The main responsibility of this layer is to provide or define the data format and **encryption**. The presentation layer is also called as Syntax layer since it is responsible for maintaining the proper syntax of the data which it either receives or transmits to other layer(s).

Application – this layer is the closest to the user. It enables network applications to communicate with other network applications.

Protocols and layers:

Application	HTTP
Presentation	MIME
Session	SSL, NetBIOS
Transport	TCP, UDP
Network	IP, ICMP
Data Link	PPP, HDLC
Physical	Ethernet

▼ TCP/IP model

It consists of four layers: Network Access, Internet, Transport, and Application.

Application
Transport
Internet
Network Access

The following picture show the comparison between the TCP/IP model and OSI model:

- The OSI Model is a logical and conceptual model that defines **network communication used by systems** open to interconnection and communication with other systems.

  On the other hand, TCP/IP helps you to determine how a **specific computer** should be connected to the internet and how you can be transmitted between them.

- OSI model helps you to standardize router, switch, motherboard, and other hardware whereas TCP/IP helps you to establish a connection between different types of computers.

TCP/IP model	OSI model
Application	Application
	Presentation
	Session
Transport	Transport
Internet	Network
Network Access	Data Link
	Physical

TCP/IP Suite of protocols

The TCP/IP suite is a set of protocols used on computer networks today (most notably on the Internet). It provides an end-to-end connectivity by specifying how data should be packetized, addressed, transmitted, routed and received on a TCP/IP network. This functionality is organised into four abstraction layers and each protocol in the suite resides in a particular layer.

The TCP/IP suite is named after its most important protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). Some of the protocols included in the TCP/IP suite are:

- **ARP (Address Resolution Protocol)** – It is responsible to find the hardware address of a host from a know IP address.
  Most of the computer programs/applications use logical address (IP address) to send/receive messages, however the actual communication happens over the physical address (MAC address) i.e from layer 2 of OSI model. So our mission is to **get the destination MAC address** which helps in communicating with other devices.
- **IP (Internet Protocol)** – used to deliver packets from the source host to the destination host based on the IP addresses.
- **ICMP (Internet Control Message Protocol)** – used to detects and reports network error conditions. Used in ping.
- **TCP (Transmission Control Protocol)** – a connection-oriented protocol that enables reliable data transfer between two computers.
- **UDP (User Datagram Protocol)** – a connectionless protocol for data transfer. Since a session is not created before the data transfer, there is no guarantee of data delivery.
- **FTP (File Transfer Protocol)** – used for file transfers from one host to another.
- **Telnet (Telecommunications Network)** – used to connect and issue commands on a remote computer.
- **DNS (Domain Name System)** – used for host names to the IP address resolution. Basically, it translates human readable **domain names** (for example, www.amazon.com) **to machine readable IP** addresses (for example, 192.0.2.44).

  ▼ How Does **DNS Route Traffic To Your Web Application**?

The following diagram gives an overview of how recursive and authoritative DNS services work together to route an end user to your website or application.

1. A user opens a web browser, enters www.example.com in the address bar, and presses Enter.

2. The request for www.example.com is routed to a DNS resolver, which is typically managed by the user's Internet service provider (ISP), such as a cable Internet provider, a DSL broadband provider, or a corporate network.

3. The DNS resolver for the ISP forwards the request for www.example.com to a DNS root name server.

4. The DNS resolver for the ISP forwards the request for www.example.com again, this time to one of the TLD name servers for .com domains. The name server for .com domains responds to the request with the names of the four Amazon Route 53 name servers that are associated with the example.com domain.

A TLD (**top-level domain**) is the most generic domain in the Internet's hierarchical DNS (domain name system). A TLD is the final component of a domain name, for example, "org" in developer.mozilla.org.

5. The DNS resolver for the ISP chooses an **Amazon Route 53** name server and forwards the request for www.example.com to that name server.

6. The Amazon Route 53 name server looks in the example.com hosted zone for the www.example.com record, gets the associated value, such as the IP address for a web server, 192.0.2.44, and returns the IP address to the DNS resolver.

Q. **Do some websites have the same IP address?**

A single server can host multiple sites.

7. The DNS resolver for the ISP finally has the IP address that the user needs. The resolver returns that value to the web browser. The DNS resolver also caches (stores) the IP address for example.com for an amount of time that you specify so that it can respond more quickly the next time someone browses to example.com. For more information, see time to live (TTL).

8. The web browser sends a request for www.example.com to the IP address that it got from the DNS resolver. This is where your content is, for example, a web server running on an Amazon EC2 instance or an Amazon S3 bucket that's configured as a website endpoint.

9. The web server or other resource at 192.0.2.44 returns the web page for www.example.com to the web browser, and the web browser displays the page.

- **HTTP (Hypertext Transfer Protocol)** – HTTP is a protocol for fetching resources such as **HTML documents**. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. Standard **TCP port 80 used by http and TCP port 443 is used by https**.

The following table shows which protocols reside on which layer of the TCP/IP model:

Layer	Protocol
Application	HTTP, NFS, DNS, telnet, FTP, SNMP
Transport	TCP, UDP
Internet	IPv4, IPv6, ARP, ICMP
Link	Ethernet (IEEE 802.3), Token Ring, FDDI

Switch operates on layer 2 - data link layer concerned with the mac addresses.

Bridges are similar to switches which also works on the layer 2.

Router operates on layer 3 - network layer concerned with the ip address

Hub basically repeats the electrical signals (it is dumb!). It works on the physical layer i.e. layer 1.

▼ Encapsulation

The term **encapsulation** is used to describe a process of adding headers and trailers around some data.

For example, here is what happens when you send an email using your favourite email program (such as Outlook or Thunderbird). Taken in reference TCP/IP model:

1. the email is sent from the Application layer to the Transport layer.
2. the Transport layer encapsulates the data and adds its own header with its own information, such as which port will be used and passes the data to the Internet layer
3. the Internet layer encapsulates the received data and adds its own header, usually with information about the source and destination IP addresses. The Internet layer than passes the data to the Network Access layer
4. the **Network Access layer is the only layer that adds both a header and a trailer**. The data is then sent through a physical network link.

Here is a graphical representation of how each layer add its own information:

Frame header	IP header	TCP header	Data	Frame trailer

Similar for OSI model:

Just like with the TCP/IP layers, each OSI layer asks for services from the next lower layer. The lower layer encapsulates the higher layer's data between a header (Data Link protocols also add a trailer).

While the TCP/IP model uses terms like segment, packet and frame to refer to a data packet defined by a particular layer, the OSI model uses a different term: **protocol data unit (PDU)**. A PDU represent a unit of data with headers and trailers for the particular layer, as well as the encapsulated data. Since the OSI model has 7 layers, PDUs are numbered from 1 to 7, with the Physical layer being the first one. For example, the term Layer 3 PDU refers to the data encapsulated at the Network layer of the OSI model.

Here is a graphical representation of all the PDUs in the OSI model:

▼ Transport layer

▼ What is transport layer?

The upper layers, the Application Layer, Presentation Layer, and Session Layer, are responsible for preparing and **sending the raw data**. In contrast, the lower layers, the Network Layer, Data Link Layer, and Physical Layer, are responsible for **encapsulating the raw data** by using headers so that the network devices like routers and switches can understand and direct the traffic to the right device.

Networking devices, like routers and switches, and end devices, like desktops and servers, have **limitations on the amount of data** that can be inserted in an **IP packet**. Because of that, the Transport Layer **segments and reassemble the data** (messages) between the sender and the receiver.

Whenever the hosts send a message into the network (internet), the Transport Layer prepares and separates the raw data (message) into smaller pieces of data for delivery. When received on the other hosts, the Transport Layer reassembles those smaller pieces of data and sends them to the upper layers.

The Application Layer has a lot of protocols that recognize the function of each data. Email traffic uses SMTP and POP3 protocols, while HTTP and HTTPS are the protocols used for web browsing. Each protocol is formatted differently based on its purpose. If a different protocol is received on a specific application, then the application will drop the data. If the web server is receiving an SMTP protocol, then the data will be dropped as the web server is expecting to receive an HTTP or HTTPS protocol. **The role of the Transport Layer is to ensure that the data is transmitted and delivered to the intended application.**

Transport layer Protocols - TCP and UDP

The commonly used Transport Layer protocols responsible for message delivery are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

TCP is a connection-oriented protocol which means it guarantees the delivery of the message, while UDP is a connectionless protocol that sends the data without

error correction. Under the TCP and UDP are **port numbers that are used to distinguish the specific type of application**. A specific port number is attached when sending the data so that the data will be received exactly to the intended application. The below diagram shows a segment in which the raw data is encapsulated by transport header (source and destination port).

▼ Ports

A **port** is a 16-bit number [0 to 65535 ports] used to identify specific applications and services. TCP and UDP specify the source and destination port numbers in their packet headers and that information, along with the source and destination IP addresses and the transport protocol (TCP or UDP), enables applications running on hosts on a TCP/IP network to communicate.

Applications that provide a service (such as **FTP and HTTP servers**) open a port on the local computer and listen for connection requests.

> A client can request the service by pointing the request to the application's IP address and port. A client can use any locally unused port number for communication.

Consider the following example:

In the picture above you can see that a host with an IP address of 192.168.0.50 wants to communicate with the FTP server. Because FTP servers use, by default, **the well-known port 21**, the host generates the request and sends it to the FTP server's IP address and port. The **host use the locally unused port** of 1200 for communication. The FTP server receives the request, generates the response,and sends it to the host's IP address and port.

The first 1024 ports are reserved for use by certain privileged services:

TCP		UDP	
FTP	20,21	DNS	53
SSH	22	BooTPS/DHCP	67
Telnet	23	TFTP	69
SMTP	25	NTP	123
DNS	53	SNMP	161
HTTP	80		
POP3	110		
IMAP4	143		
HTTPS	443		

NOTE: The combination of an IP address(layer 3) and a port(layer 4) number is called a **socket (layer 5)**. In our example the socket would be 192.168.0.50:1200.

What's the difference between POP and IMAP? POP3 downloads the email from a server to a single computer, then deletes the email from the server. On the other hand, **IMAP stores the message on a server and synchronizes the message across multiple devices**

.

Type of ports

1. **Ports 0-1023 – Well known ports**

   These are allocated to server services by the Internet Assigned Numbers Authority (IANA). e.g Web servers normally use port 80 and SMTP servers use port 25.

2. **Ports 1024-49151 – Registered Port**

   These can be registered for services with the IANA and should be treated as semi-reserved. User written programs should not use these ports.

3. **Ports 49152-65535** – **Ephemeral ports**

   You are free to use these in **client programs**. When a Web browser connects to a web server the browser will allocate itself a port in this range.

   An **ephemeral port** is a communications endpoint (port) of a transport layer protocol of the Internet protocol suite that is used for only a short period of time for the duration of a communication session. Such short-lived ports are allocated automatically within a predefined range of port numbers by the IP stack software of a computer operating system. TCP, UDP and the Stream Control Transmission Protocol (SCTP) typically use an ephemeral port for the client-end of a client–server communication. At the server end of the communication session, ephemeral ports may also be used for continuation of communications with a client that initially connected to one of the services listening with a well-known port. For example, the Trivial File Transfer Protocol (TFTP) and Remote Procedure Call (RPC) applications can behave in this manner.

   The allocation of an ephemeral port is temporary and only valid for the duration of the communication session. After completion of the session, the port is destroyed and the port number becomes available for reuse, but many

implementations simply increment the last used port number until the ephemeral port range is exhausted, when the numbers roll over. Ephemeral ports are also called ***dynamic ports***, because they are used on a per request basis, and are only known by number once allocated.

▼ TCP explained

One of the main protocols in the TCP/IP suite is Transmission Control Protocol (TCP). TCP provides reliable and ordered delivery of data between applications running on hosts on a TCP/IP network. Because of its **reliable nature**, TCP is used by applications that require high reliability, such as FTP, SSH, SMTP, HTTP, etc.

TCP is connection-oriented, which means that, before data is sent, a connection between two hosts must be established. The process used to establish a TCP connection is known as the three-way handshake. **After the connection has been established, the data transfer phase begins.** After the data is transmitted, the connection is terminated.

One other notable characteristic of TCP is its reliable delivery. TCP uses sequence numbers to identify the order of the bytes sent from each computer so that the data can be reconstructed in order. If any data is lost during the transmission, the sender can retransmit the data.

Byte streaming: Application layer continuously sends data to the transport layer. TCP carries segmentation which means it converts the continuous data coming to transport layer into segments. Each segments is a collection of bytes. This segmentation is necessary so that the information can be placed inside the TCP data field.

Because of all of its characteristics, TCP is considered to be complicated and costly in terms of network usage. The TCP header is up to **24 bytes** long and consists of the following fields:

source port (16 bits)	destination port (16 bits)
sequence number (32 bits)	
acknowledgment number (32 bits)	
header length (4 bit)	reserved
flags	window (16 bits)
checksum (16 bit)	urgent (16 bits)
options	

- **source port** – the port number of the application on the host sending the data.
- **destination port** – the port number of the application on the host receiving the data.
- **sequence number** – used to identify each byte of data.
- **acknowledgment number** – the next sequence number that the receiver is expecting.
- **header length** – the size of the TCP header.
- **reserved** – always set to 0.
- **flags** – used to set up and terminate a session.
- **window** – the window size the sender is willing to accept.
- **checksum** – used for error-checking of the header and data.
- **urgent** – indicates the offset from the current sequence number, where the segment of non-urgent data begins.
- **options** – various TCP options, such as Maximum Segment Size (MSS) or Window Scaling.

▼ TCP three-way handshake

Since TCP is a connection-oriented protocol, a connection needs to be established before two devices can communicate. TCP uses a process called three-way handshake to negotiate the sequence and acknowledgment fields and start the session. Here is a graphical representation of the process:

As the name implies, the three way handshake process consists of three steps:

1. Host A initiates the connection by sending the TCP SYN (synchronize) packet to the destination host. The packet contains the random sequence number (e.g. **5432**) which marks the beginning of the sequence numbers for data that the Host A will transmit.
2. The Server receives the packet and responds with its own sequence number. The response also includes the acknowledgment number, which is Host A's sequence number incremented by 1 (in our case, that would be **5433**).
3. Host A acknowledges the response of the Server by sending the acknowledgment number, which is the Server's sequence number incremented by 1.

Here is another picture with the numbers included:

After the data transmission process is finished, TCP will terminate the connection between two endpoints. This four-step process is illustrated below:

1. The client application that wants to close the connection sends a TCP segment with the FIN (**Finished**) flag set to 1.

2. The server receives the TCP segment and acknowledges it with the ACK segment.
3. Server sends its own TCP segment with the FIN flag set to 1 to the client in order to terminate the connection.
4. The client acknowledges the server's FIN segment and closes the connection.

▼ UDP explained

Because it does not provide many features that TCP does, UDP uses much less network resources than TCP. UDP is commonly used with two types of applications:

One other important protocol in the TCP/IP site is User Datagram Protocol (UDP). This protocol is basically a scaled-down version of TCP. Just like TCP, this protocol **provides delivery of data between applications** running on hosts on a TCP/IP network, but, unlike TCP, it does not sequence the data and does not care about the order in which the segments arrive at the destination. Because of this, it is considered to be an unreliable protocol. UDP is also considered to be a connectionless protocol since **no virtual circuit is established between two endpoints** before the data transfer takes place.

- **applications that are tolerant of the lost data** – VoIP (Voice over IP) uses UDP because if a voice packet is lost, by the time the packet would be retransmitted, too much delay would have occurred, and the voice would be unintelligible.
- **applications that have some application mechanism to recover lost data** – Network File System (NFS) performs recovery with application layer code, so UDP is used as a transport-layer protocol.

The UDP header is **8 bytes** long and consists of the following fields:

source port (16 bits)	destination port (16 bits)
length (16 bits)	checksum (16 bits)

Here is a description of each field:

- **source port** – the port number of the application on the host sending the data.

- **destination port** – the port number of the application on the host receiving the data.
- **length** – the length of the UDP header and data.
- **checksum** – checksum of both the UDP header and UDP data fields.

▼ Network layer

▼ IP header

An **IP header** is a prefix to an IP packet that contains information about the IP version, length of the packet, **source and destination IP addresses**, etc. It consists of the following fields:

Version (4 bits)	Header length (4 bits)	Priority and Type of Service (8 bits)	Total length (16 bits)
Identification (16 bits)		Flags (3 bits)	Fragmented offset (13 bits)
Time to live (8 bits)	Protocol (8 bits)	Header checksum (16 bits)	
Source IP address (32 bits)			
Destination IP address (32 bits)			
Options (up to 32 bits)			

Here is a description of each field:

- **Version** – the version of the IP protocol. For IPv4, this field has a value of 4.
- **Header length** – the length of the header in 32-bit words. The minumum value is 20 bytes, and the maximum value is 60 bytes.
- **Priority and Type of Service** – specifies how the datagram should be handled. The first 3 bits are the priority bits.
- **Total length** – the length of the entire packet (header + data). The minimum length is 20 bytes, and the maximum is 65,535 bytes.
- **Identification** – used to differentiate fragmented packets from different datagrams.
- **Flags** – used to control or identify fragments.
- **Fragmented offset** – used for fragmentation and reassembly if the packet is too large to put in a frame.
- **Time to live** – limits a datagram's lifetime. If the packet doesn't get to its destination before the TTL expires, it is discarded.

- **Protocol** – defines the protocol used in the data portion of the IP datagram. For example, TCP is represented by the number 6 and UDP by 17.
- **Header checksum** – used for error-checking of the header. If a packet arrives at a router and the router calculates a different checksum than the one specified in this field, the packet will be discarded.
- **Source IP address** – the IP address of the host that sent the packet.
- **Destination IP address** – the IP address of the host that should receive the packet.
- **Options** – used for network testing, debugging, security, and more. This field is usually empty.

Consider the following IP header, captured with **Wireshark**:

```
⊟ Internet Protocol Version 4, Src: 192.168.5.45 (192.168.5.45), Dst: 91.198.174.192 (91.198.174.192)
    Version: 4
    Header Length: 20 bytes
  ⊟ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
    Total Length: 52
    Identification: 0x4116 (16662)
  ⊟ Flags: 0x02 (Don't Fragment)
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
  ⊟ Header checksum: 0x0000 [validation disabled]
      [Good: False]
      [Bad: False]
    Source: 192.168.5.45 (192.168.5.45)
    Destination: 91.198.174.192 (91.198.174.192)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
```

Notice the fields in the header: the IP version is IPv4, the header length is 20 bytes, the upper-level protocol used is TCP, the TTL [**Time-to-live** (TTL) is a value for the period of time that a packet, or data, should exist on a computer or network before being discarded. Denoted in seconds] value is set to 128, source and destination IP addresses are listed, etc.

▼ Unicast, multicast, and broadcast addresses

There are three types of Ethernet addresses:

1. unicast addresses

Unicast addresses represent a single LAN interface. A unicast frame will be sent to a specific device, not to a group of devices on the LAN:

The unicast address will have the value of the MAC address of the destination device.

2. multicast addresses

Multicast addresses represent a **group of devices** (not all as in broadcast) in a LAN. A frame sent to a multicast address will be forwarded to a group of devices on the LAN:

Multicast frames (frames is called in data link layer) have a value of **1 in the least-significant bit** of the first octet of the destination address. This helps a network switch to distinguish between unicast and multicast addresses. One example of an Ethernet multicast address would be **01:00:0C:CC:CC:CC**, which is the address used by CDP (Cisco Discovery Protocol).

IPv4 → class D. [read next topic to know process]

IPv6 → FF00::18

3. broadcast addresses

Broadcast addresses represent all device on the LAN. Frames sent to a broadcast address will be delivered to all devices on the LAN:

The **limited broadcast** address is defined as which sends or broadcast packet to **all hosts** connected to its own or home network. Using 255.255.255.255 or 0.0.0.0

The **Directed Broadcast** IP address is what is known as the Broadcast IP for each **Subnet**. Host 1 has the IP address `10.1.1.11` on the `10.1.1.0/24` network. The Broadcast IP address of this IP subnet is therefore `10.1.1.255`. The switch will flood broadcast frames out all ports except the port that it was received on.

NOTE: IPv6 does not have any broadcast address.

▼ Types of IP addresses

The IP addresses are divided into three different types, based on their operational characteristics:

1. unicast IP addresses – an address of a single interface. The IP addresses of this type are used for one-to-one communication. Unicast IP addresses are used to direct packets to a specific host. Here is an example:

In the picture above you can see that the host wants to communicate with the server. It uses the (unicast) IP address of the server (**192.168.0.150**) to do so.

2. multicast IP addresses – used for one-to-many communication. Multicast messages are sent to IP multicast group addresses. Routers forward copies of the packet out to every interface that has **hosts subscribed to that group address**. Only the hosts that need to receive the message will process the packets. All other hosts on the LAN will discard them. Here is an example:

R1 has sent a multicast packet destined for **224.0.0.9**. This is an RIPv2 packet, and only routers on the network should read it. R2 will receive the packet and read it. All other hosts on the LAN will discard the packet.

Eg. Used in video conferencing. A class D address is used let say 239.1.1.1 and we will start sending data to that particular class D address. The receiver will join that particular group using protocols like IGMP and then they start receiving the multicast data.

3. broadcast IP addresses – used to send data to all possible destinations in the broadcast domain (the one-to-everybody communication). The broadcast address for a network has all host bits on. For example, for the network **192.168.30.0** the broadcast address would be **192.168.30.255***. Also, the IP address of all 1's (**255.255.255.255**) can be used for local broadcast. Here's an example:

R1 wants to communicate with all hosts on the network and has sent a broadcast packet to the broadcast IP address of 192.168.30.255. All hosts in the same broadcast domain will receive and process the packet.

▼ IPv4 Address

IP is the principal set of rules for communication on the Internet. An IP address is needed to be allocated on the devices, such as PCs, printers, servers, routers, switches, etc., to be able to communicate with each other in the network and out the Internet.

IPv4 Address Format

Each set of IPv4 address is called an 'octet' because a set is composed of 8 bits. The figure below shows the binary format of each octet in the 192.168.10.100 IP address:

Format	1st Octet	2nd Octet	3rd Octet	4th Octet
Dotted Decimal	192	168	10	100
Binary	1100 0000	1010 1000	0000 1010	0110 0100

A number in an octet can range from 0 to 255. Therefore, the full IPv4 address space goes from 0.0.0.0 to 255.255.255.255. The IPv4 address has two parts, the network part and the host part. A **subnet mask** is used to identify these parts.

Network Part

The network part of the IPv4 address is on the left-hand side of the IP address. It specifies the particular network to where the IPv4 address belongs. The network portion of the address also identifies the IP address class of the IPv4 address.

For example, we have the IPv4 address 192.168.10.100 and **a /24 subnet mask. /24 simply means that the first 24 bits, starting from the left side, is the network portion** of the IPv4 address. The 8 remaining bits of the 32 bits will be the host portion.

Format	1st Octet	2nd Octet	3rd Octet	4th Octet
Dotted Decimal	192	168	10	100
Binary	1100 0000	1010 1000	0000 1010	0110 0100

Host Part

The host portion of the IPv4 address **uniquely identifies the device or the interface on your network**. Hosts that have the same network portion can communicate with one another directly, without the need for the traffic to be routed.

IPv4 Address Allocation

The Internet Protocol address can be allocated to hosts or interfaces either manually or dynamically.

- **Static** – static IP address is set manually on the device. It is best practice to set static IP addresses on network devices, such as routers and switches, and on servers as well.
- **Dynamic** – dynamic IP address can be automatically allocated to a device via Dynamic Host Configuration Protocol (DHCP). Dynamic IP addresses are best

to be used on end devices, such as PCs.

Types of IPv4 Addresses

We have two types of IP addresses, namely public IP addresses and private IP addresses.

- **Public IP address** – used to route Internet traffic. This is used on the Internet and is given out by Internet Service Providers (ISPs) to their customers.
- **Private IP address** – used in private networks for internal traffics within the LAN. Private addresses are not routable out the Internet.

▼ Non routing IP address

Billions of people are on the internet, and its growth is exponential. With a maximum of 4 billion available addresses, the size of IPv4 address space is fast depleting. Even though IPv6 has umpteen advantages, due to many practical reasons, it cannot supplant IPv4. Therefore, both versions must co-exist for some time for seamless migration.

However, RFC 1918 was released in 1996 to counter the IPv4 depletion. It quotes that “there can be networks within an autonomous system known as Non-Routable Address Space. These defined IP ranges can build the private or local network”.

There are three different IP ranges defined in RFC 1918. They are:

- 10.0.0.0/8 (Range: 10.0.0.0 – 10.255.255.255) – Available IPs: 16,777,214
- 172.16.0.0/12 (Range: 172.16.0.0 – 172.31.255.255) – Available IPs: 1,048,574
- 192.168.0.0/16 (Range: 192.168.0.0 – 192.168.255.255) – Available IPs: 65,534

Further on, these private networks connect to the internet through NAT technology

NAT acts as a middleman between the internet and private networks. It works at layer 3 of the OSI model and operates on a router. It enables a private

network that uses the non-routable IP addresses **to connect with the internet through the translated Public IP**.

▼ NAT

To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of a private IP address to a public IP address is required. **Network Address Translation (NAT)** is a process in which **one or more local IP address is translated into one or more Global IP address** and vice versa in order to provide Internet access to the local hosts. Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on a router or firewall.

One of the solution to resolve the issue of all IPs being used out.

Network Address Translation (NAT) - GeeksforGeeks

To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the

https://www.geeksforgeeks.org/network-address-translation-nat/

https://www.youtube.com/watch?v=16g5IKTJzXc

Subnet mask

An IP address is divided into two parts: network and host parts. For example, an IP class A address consists of 8 bits identifying the network and 24 bits identifying the host. This is because the default subnet mask for a class A IP address is 8 bits long. (or, written in dotted decimal notation, 255.0.0.0). What does it mean? Well, like an IP address, a subnet mask also consists of 32 bits. Computers use it to

determine the network part and the host part of an address. The 1s in the subnet mask represent a network part, the 0s a host part.

Let's say that we have the IP address of 10.0.0.1 with the default subnet mask of 8 bits (255.0.0.0). Only IP addresses that begins with 10 will be in the same network. So, in this case, the range of addresses in this network is 10.0.0.0 – 10.255.255.255.

NOTE: A subnet mask must always be a series of 1s followed by a series of 0s.

Slash notation: The number of succeeding 1s are 8, therefore the slash notation of 255.0.0.0 is /8.

▼ Classes of IPv4 addresses

TCP/IP defines five classes of IP addresses: class A, B, C, D, and E. Each class has a range of valid IP addresses. **The value of the first octet determines the class**. IP addresses from the first three classes (A, B and C) can be used for **host addresses**. The other two classes are used for other purposes – class D for **multicast** and class E for **experimental purposes**.

The system of IP address classes was developed for the purpose of Internet IP addresses assignment. The classes created were based on the network size. For example, for the small number of networks with a very large number of hosts, the Class A was created. The Class C was created for numerous networks with small number of hosts.

Classes of IP addresses are:

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

- A - 00000000 to 01111111
- B - 10000000 to 10111111
- C - 11000000 to 11011111

- D - 11100000 to 11101111
- E - 11110000 to 11111111

For the IP addresses from Class A, the first 8 bits (the first decimal number) represent the network part, while the remaining 24 bits represent the host part. For Class B, the first 16 bits (the first two numbers) represent the network part, while the remaining 16 bits represent the host part. For Class C, the first 24 bits represent the network part, while the remaining 8 bits represent the host part.

Consider the following IP addresses:

- **10.50.120.7** – because this is a Class A address, the first number (10) represents the network part, while the remainder of the address represents the host part (50.120.7). This means that, in order for devices to be on the same network, the first number of their IP addresses has to be the same for both devices. In this case, a device with the IP address of 10.47.8.4 is on the same network as the device with the IP address listed above. The device with the IP address **11.5.4.3 is not on the same network**, because the first number of its IP address is different.
- **172.16.55.13** – because this is a Class B address, the first two numbers (172.16) represent the network part, while the remainder of the address represents the host part (55.13). A device with the IP address of 172.16.254.3 is on the same network, while a device with the IP address of 172.55.54.74 isn't.

NOTE: The system of network address ranges described here is generally bypassed today by use of the Classless Inter-Domain Routing (CIDR) addressing.

Special IP address ranges that are used for special purposes are:

- **0.0.0.0/8** – addresses used to communicate with the local network
- **127.0.0.0/8** – loopback addresses
- **169.254.0.0/16** – link-local addresses (APIPA)

▼ DHCP

DHCP (Dynamic Host Configuration Protocol)

DHCP is a **network protocol** that is used to assign various network parameters to a device. This greatly simplifies administration of a network, since there is no need to assign static network parameters for each device.

DHCP is a client-server protocol. A client is a device that is configured to use DHCP to request network parameters from a DHCP server. **DHCP server maintains a pool of available IP addresses and assignes one of them to the host.** A DHCP server can also provide some other parameters, such as:

- subnet mask
- default gateway
- domain name
- DNS server

Cisco routers can be configured as both DHCP client and DHCP server.

DHCP process explained:

DHCP client goes through the four step process:

1: A DHCP client sends a broadcast packet (**DHCP Discover**) to discover DHCP servers on the LAN segment.

2: The DHCP servers receive the DHCP Discover packet and respond with **DHCP Offer** packets, offering IP addressing information.

3: If the client receives the DHCP Offer packets from multiple DHCP servers, the first DHCP Offer packet is accepted. The client responds by broadcasting a **DHCP Request** packet, requesting the network parameters from the server that responded first.

4: The DHCP server approves the lease with a **DHCP Acknowledgement** packet. The packet includes the lease duration and other configuration information.

NOTE: DHCP uses a well-known UDP port number 67 for the DHCP server, and the UDP port number 68 for the client.

▼ Subnetting

▼ What is subnetting?

Subnetting is the practice of dividing a network into two or more smaller networks. It increases routing efficiency, enhances the security of the network and reduces the size of the broadcast domain.

Consider the following example:

In the picture above we have one huge network: **10.0.0.0/24**. All hosts on the network are in the same subnet, which has the following disadvantages:

- **a single broadcast domain** – all hosts are in the same broadcast domain. A broadcast sent by any device on the network will be processed by all hosts, creating lots of unnecessary traffic.
- **network security** – each device can reach any other device on the network, which can present security problems. For example, a server containing sensitive information shouldn't be in the same network as user's workstations.
- **organizational problems** – in a large networks, different departments are usually grouped into different subnets. For example, you can group all devices from the **Accounting** department in the same subnet and then give access to sensitive financial data only to hosts from that subnet.

The network above could be subnetted like this:

Now, two subnets were created for different departments: **10.0.0.0/24** for Accounting and **10.1.0.0/24** for Marketing. Devices in each subnet are now in a

different broadcast domain. This will reduce the amount of traffic flowing on the network and allow us to implement packet filtering on the router.

▼ CIDR (Classless inter-domain routing)

▼ What is CIDR?

CIDR (Classless inter-domain routing) is a method of **public IP** address assignment. It was introduced in 1993 by Internet Engineering Task Force with the following goals:

- to deal with the IPv4 address exhaustion problem
- to slow down the growth of routing tables on Internet routers

Before CIDR, public IP addresses were assigned based on the class boundaries:

- **Class A** – the classful subnet mask is /8. The number of possible IP addresses is 16,777,216 (2 to the power of 24).
- **Class B** – the classful subnet mask is /16. The number of addresses is 65,536
- **Class C** – the classful subnet mask is /24. Only 256 addresses available.

Some organisations were known to have gotten an **entire Class A public IP address** (for example, IBM got all the addresses in the 9.0.0.0/8 range). Since these addresses can't be assigned to other companies, there was a shortage of available IPv4 addresses. Also, since IBM probably didn't need more than 16 million IP addresses, a l**ot of addresses were unused**.

To combat this, the classful network scheme of allocating the IP address was abandoned. The new system was classless – a classful network was split into multiple smaller networks. For example, if a company needs 12 public IP addresses, it would get something like this: **190.5.4.16/28**.

The number of usable IP addresses can be calculated with the following formula:

2 to the power of host bits – 2

In the example above, the company got 14 usable IP addresses from the 190.5.4.16 – 190.5.4.32 range because there are 4 host bits and 2 to the power of 4 minus 2 is 14. The **first and the last address are the network address and the broadcast address**,respectively. All other addresses inside the range could be assigned to Internet hosts.

▼ Create subnets

There are a couple of ways to create subnets. In this article we will subnet a class C address **192.168.0.0** that, by default, has **24** subnet bits and **8** host bits.

Before we start subnetting, we have to ask ourselves these two questions:

1. How many subnets do we need?

2x = number of subnets. **x** is the number of 1s in the subnet mask. With 1 subnet bit, we can have 21 or 2 subnets. With 2 bits, 22 or 4 subnets, with 3 bits, 23 or 8 subnets, etc.

2. How many hosts per subnet do we need?

2y – 2 = number of hosts per subnet. **y** is the number of 0s in the subnet mask.

Subnetting example

An example will help you understand the subnetting concept. Let's say that we need to subnet a class C address **192.168.0.0/24**. We need two subnets with **50** hosts per subnet. Here is our calculation:

1. Since we need only two subnets, we need 21 subnet bits. In our case, this means that we will take one bit from the host part. Here is the calculation:

First, we have a class C address 192.168.0.0 with the subnet mask of 24. Let's convert them to binary:

192.168.0.0 = 11000000.10101000.00000000.00000000

255.255.255.0 = 11111111.11111111.11111111.00000000

We need to take a single zero from the host part of the subnet mask. Here is our new subnet mask:

255.255.255.128 = 11111111.11111111.11111111.10000000

Remember, the ones in the subnet mask represent the network.

2. We need 50 hosts per subnet. Since we took one bit from the host part, we are left with seven bits for the hosts. Is it enough for 50 hosts? The formula to calculate the number of hosts is 2y – 2, with **y** representing the number of host bits. Since 27 – 2 is 126, we have more than enough bits for our hosts.

3. Our network will look like this:

192.168.0.0/25 – the first subnet has the subnet number of **192.168.0.0**. The range of IP addresses in this subnet is **192.168.0.0 – 192.168.0.127**.

192.168.0.128/25 – the second subnet has the subnet number of **192.168.0.128**. The range of IP addresses in this subnet is **192.168.0.128 – 192.168.0.255**.

▼ Private IP addresses explained

The original design of the Internet intended that each host on every network should have a real, routable IP address. An organization that would like to access the Internet would complete some paperwork to describe its internal network and the number of hosts on it. The organization would then receive a number of IP addresses, according to its needs. But there was one huge problem with this concept – if each host on each network in the world was provided with an unique IP address, we would have run out of IP addresses a long time ago!

Therefore, the concept of private IP addressing was developed to address the IP address exhaustion problem. The **private IP addresses can be used on the private network of any organization in the world and are not globally unique.**

Consider the following example:

In the example above you can see that two unrelated organizations use the same private IP network (10.0.0.0/24) inside their respective internal networks. Because private IP addresses are not globally unique, both organizations can use private IP addresses from the same range. **To access the Internet, the organizations can use a technology called Network Address Translation** (NAT), which we will describe in the later lessons.

There are three ranges of addresses that can be used in a private network (e.g. your home LAN or office)

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.**0.0 –** 192.168.**255.255**

Internet routers are configured to discard any packets coming from the private IP address ranges, so these addresses are not routable on the Internet.

▼ Data-Link Layer

▼ Ethernet explained

Ethernet is the most used networking technology for LANs today. It defines wiring and signaling for the Physical layer of the OSI model. For the Data Link layer, it defines frame formats and protocols.

Ethernet is described as **IEEE 802.3** standard. It uses **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)** access method and supports speeds up to 100 Gbps. It can use coaxial, twisted pair and fiber optic cables. Ethernet uses frames to with source and destination MAC addresses to deliver data.

NOTE: The term **Ethernet LAN** refers to a combination of computers, switches, and different kinds of cables that use the Ethernet standard to communicate over the network. It is by far the most popular LAN technology today.

▼ Ethernet frame

We have already learned that encapsulated data defined by the Network Access layer is called an Ethernet frame. An Ethernet frame starts with a header, which contains the source and destination MAC addresses, among other data. The middle part of the frame is the actual data. The frame ends with a field called Frame Check Sequence (FCS).

The Ethernet frame structure is defined in the IEEE 802.3 standard. Here is a graphical representation of an Ethernet frame and a description of each field in the frame:

Preamble	SFD	Destination MAC	Source MAC	Type	Data and Pad	FCS
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46-1500 Bytes	4 Bytes

- **Preamble** – informs the receiving system that a frame is starting and enables synchronisation.
- **SFD (Start Frame Delimiter)** – signifies that the Destination MAC Address field begins with the next byte.
- **Destination MAC** – identifies the receiving system.
- **Source MAC** – identifies the sending system.
- **Type** – defines the type of protocol inside the frame, for example IPv4 or IPv6.

- **Data and Pad** – contains the payload data. Padding data is added to meet the minimum length requirement for this field (46 bytes).
- **FCS (Frame Check Sequence)** – contains a 32-bit Cyclic Redundancy Check (CRC) which allows detection of corrupted data.

The FCS field is the only field present in the Ethernet trailer. It allows the receiver to discover whether errors occurred in the frame. **Note that Ethernet only detects in-transit corruption of data – it does not attempt to recover a lost frame. Other higher level protocols (e.g. TCP) perform error recovery.**

▼ MAC & IP address

▼ MAC address

A Media Access Control (MAC) address is a **48-bit** (6 bytes) address that is used for **communication between two hosts in an Ethernet environment**. It is a hardware address, which means that it is stored in the firmware of the network card.

Every network card manufacturer gets a universally unique 3-byte code called the Organizationally Unique Identifier (OUI). Manufacturers agree to give all NICs (network interface card) a MAC address that begins with the assigned OUI. The manufacturer then assigns a unique value for the last 3 bytes, which ensures that every MAC address is globaly unique.

MAC addresses are usually written in the form of 12 hexadecimal digits. For example, consider the following MAC address:

```
D8-D3-85-EB-12-E3
```

Every hexadecimal character represents 4 bits, so the **first six hexadecimal characters represent the vendor** (Hewlett Packard in this case).

How to find out your own MAC address?

Type the *ipconfig/all* command in cmd and you should see a field called **Physical Address** under the Ethernet adapter settings:

```
Command Prompt

C:\Users\user>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : WIN-7NHASVKCI7D
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : localdomain

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-6C-F3-E5
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::b82d:1e2b:ed4d:b89d%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.10.100.131(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, March 25, 2013 2:34:36 PM
   Lease Expires . . . . . . . . . . : Monday, March 25, 2013 3:04:36 PM
   Default Gateway . . . . . . . . . :
   DHCP Server . . . . . . . . . . . : 10.10.100.254
   DHCPv6 IAID . . . . . . . . . . . : 234884137
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-18-C6-CD-56-00-0C-29-6C-F3-E5

   DNS Servers . . . . . . . . . . . : 10.10.100.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter isatap.localdomain:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes

C:\Users\user>
```

If you are using Linux, type the *ifconfig* command. ether gives the MAC.

Here, *enp1s0* means **ethernet network peripheral** while *wlp2s0* means **wireless interface.** On a computer with both a wired and a wireless connection (many laptops and notebooks), there will be one MAC Address for the wired (Ethernet) connection and a second MAC for the wireless connection.

```
khush@khush-HP-ProBook-430-G3:~$ ifconfig
enp1s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 98:e7:f4:54:af:16  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 13131  bytes 1571143 (1.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13131  bytes 1571143 (1.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.164  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::1945:5e8e:107f:779b  prefixlen 64  scopeid 0x20<link>
        inet6 fd01::d17c:e1aa:75e4:4fd9  prefixlen 64  scopeid 0x0<global>
        inet6 fd01::3863:73c4:b3c9:50b8  prefixlen 64  scopeid 0x0<global>
        ether e4:a7:a0:a8:e4:4f  txqueuelen 1000  (Ethernet)
        RX packets 1601743  bytes 1149818824 (1.1 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 702071  bytes 212650282 (212.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Can also find both ethernet and wlan MAC addresses using `lshw -c net`.

▼ IP address

An IP address is a 32-bit number that identifies a host on a network. Each device that wants to communicate with other devices on a TCP/IP network needs to have an IP address configured. For example, in order to access the Internet, your computer will need to have an IP address assigned (**usually obtained by your router from the ISP**).

An IP address is usually written in the form of four decimal numbers seperated by periods (e.g. 10.0.50.1). The first part of the address represents the network the device is on (e.g. 10.0.0.0), while the second part of the address identifies the host device (e.g. 10.0.50.1).

In contrast to MAC address, an IP address is a logical address. It can be configured manually or it can be obtained from a DHCP server.

NOTE: The term *IP address* is usually used for IPv4, which is the fourth version of the IP protocol. A newer version exists, IPv6, and uses 128-bit addressing.

Private IP addresses

There are three ranges of addresses that can be used in a private network (e.g. your home LAN). These addresses are not routable through the Internet.

Private addresses ranges are:

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

How to find out your IP address

If you are using Windows, start the Command Prompt (Start – Programs – Accessories – Command Prompt). Enter the *ipconfig* command. You should see a field called **IP Address:**

Linux users:

Enter *ifconfig*. You should see a field called **inet addr** in **wlps20** (see MAC screenshot of my own pc):

My public ip address: dig +short myip.opendns.com @resolver1.opendns.com

> Got to Setting → wifi → open the wifi setting. You will get Private IPv4 and wlan MAC.

▼ WLAN vs Wifi

What is the difference between WLAN and Wi-Fi?

markrubens - Fotolia The notion of wireless continues to get complicated in business environments. If you don't know the specific wireless context or application being discussed, your

https://www.techtarget.com/searchnetworking/answer/Wireless-vs-Wi-Fi-What-is-the-difference-between-Wi-Fi-and-WLAN

▼ Physical layer

▼ What is physical layer?

Physical Layer sends data bits from one device(s) (like a computer) to another device(s). Physical Layer defines the types of encoding (that is how the 0's and 1's are encoded in a signal). Physical Layer is responsible for the communication of the unstructured raw data streams over a physical medium.

Physical Topologies :

Physical Topology or Network Topology is the Geographical Representation of Linking devices.

Star topology is by far the most common.

1. **Mesh Topology:** In a mesh topology, each and every device should have a dedicated point-to-point connection with each and every other device in the network. Here there is more security of data because there is a dedicated point-to-point connection between two devices. Mesh Topology is difficult to install because it is more complex.
2. **Star Topology:** In star topology, the device should have a dedicated point-to-point connection with a central controller or hub. Star Topology is easy to

install and reconnect as compared to Mesh Topology. Star Topology doesn't have Fault Tolerance Technique.

3. **Bus Topology:** In a bus topology, multiple devices are connected through a single cable that is known as backbone cable with the help of tap and drop lines. It is less costly as compared to Mesh Topology and Star Topology. Re-connection and Re-installation are difficult.
4. **Ring Topology:** In a ring topology, each device is connected with repeaters in a circle-like ring that's why it is called Ring Topology. In Ring topology, a device can send the data only when it has a token, without a token no device can send the data, and a token is placed by Monitor in Ring Topology.

Modes of Transmission Medium :

1. **Simplex mode:** In this mode, out of two devices, only one device can transmit the data, the other device can only receive the data. Example- Input from keyboards, monitors, TV broadcasting, Radio broadcasting, etc.
2. **Half Duplex mode:** In this mode, out of two devices, both devices can send and receive the data but only one at a time not simultaneously. Example- Walkie-Talkie, Railway Track, etc.
3. **Full-Duplex mode:** In this mode, both devices can send and receive the data simultaneously. Example- Telephone System, Chatting applications, etc.

▼ Types of ethernet cabling

There are three cable types commonly used for Ethernet cabling: coaxial, twisted pair, and fiber-optic cabling. In today's LANs, the **twisted pair cabling is the most popular** type of cabling, but the **fiber-optic cabling usage is increasing**, especially in high performance networks. Coaxial cabling is generally used for cable Internet access. Let's expain all three cable types in more detail.

Coaxial cabling

A coaxial cable has an inner conductor that runs down the middle of the cable. The conductor is surrounded by a layer of insulation which is then surrounded by another conducting shield, which makes this type of cabling resistant to outside interference. This type of cabling comes in two types – thinnet and thicknet. Both

types have maximum transmission speed of 10 Mbps. Coaxial cabling was previously used in computer networks, but today are largely replaced by twisted-pair cabling (Photo credit: Wikipedia)

Twisted-pair cabling

A twisted-pair cable has four pair of wires. These wires are twisted around each other to reduce crosstalk and outside interference. This type of cabling is common in current LANs.

Twisted-pair cabling can be used for telephone and network cabling. It comes in two versions, **UTP (Unshielded Twisted-Pair)** and **STP (Shielded Twisted-Pair)**. The difference between these two is that an STP cable has an additional layer of insulation that protects data from outside interferences.

Here you can see how a twisted pair cable looks like (Photo credit: Wikipedia):

A twisted-pair cable uses 8P8C connector, sometimes wrongly referred to as RJ45 connector (Photo credit: Wikipedia).

Fiber-optic cabling

This type of cabling uses optical fibers to transmit data in the form of light signals. The cables have strands of glass surrounded by a cladding material (Photo credit: Wikipedia):

This type of cabling can support greater cable lengths than any other cabling type (up to a couple of miles). The cables are also immune to electromagnetic interference. As you can see, this cabling method has many advantages over other methods but its main drawback is that it is more expensive.

There are two types of fiber-optic cables:

- **Single-mode fiber (SMF)** – uses only a single ray of light to carry data. Used for larger distances.
- **Multi-mode fiber (MMF)** – uses multiple rays of light to carry data. Less expensive than SMF.

Four types of connectors are commonly used:

- ST (Straight-tip connector)
- SC (Subscriber connector)
- FC (Fiber Channel)
- LC (Lucent Connector)

▼ Network devices

▼ Network devices

Let’s take a look at the network devices commonly found in today’s LANs..

Hubs

A hub serves as a central point to which all of the hosts in a network connect to. A Hub is an **OSI Layer 1** device and has no concept of Ethernet frames or

addressing. It **simply receives a signal from one port and sends it out to all other ports**. Here is an example 4-port Ethernet hub:

Today, hubs are considered obsolete and switches are commonly used instead.

Switches

Like hubs, a switch is used to connect multiple hosts together, but it has many advantages over a hub. Switch is an **OSI Layer 2** device, which means that it **can inspect received traffic and make forwarding decisions**. Each port on a switch is a separate collision domain and can run in a full duplex mode.

Routers

A router is a device that routes packets from **one network to another**. A router is most commonly an **OSI Layer 3** device. Routers divide broadcast domains and have traffic filtering capabilities.

The picture below shows a typical home router:

▼ Network hubs explained

It simply receives the signal from one port and sends it out to all other ports. It has no way of distinguishing out which port a signal should be sent to; instead, an electrical signal is sent out each port. All nodes on the network will receive data, and the data will eventually reach the correct destination, but with a **lot of unnecessary network traffic**:

In the example above you can see that the hub has sent out the receiving signal out all other ports, except the incoming port. Hubs are therefore **considered obsolete and switches are commonly used** instead in modern LANs. Hubs have numerous disadvantages over switches, such as:

- they are not aware of the traffic that passes through them
- they create only one large collision domain

- a hub typically operates in half duplex
- there is also a security issue with hubs since the traffic is forwarded to all ports (except the source port), which makes it possible to capture all traffic on a network with a network sniffer!

NOTE: Hubs are also known as **multiport repeaters** because that is basically what they do – repeat the electrical signal that comes in one port out all other ports (except the incoming port).

▼ Network switch explained

A switch manages the flow of data across a network by **inspecting the incoming frame's destination MAC address and forwarding the frame only to the host for which the data was intended**. Each switch has a dynamic table (called the MAC address table) that **maps MAC addresses to ports**. With this information, a switch can identify which system is sitting on which port and where to send the received frame.

The switching table contains MAC addresses and the switch ports on which they were learned or statically configured.

To better understand how a switch works, consider the following example:

As you can see from the example above, Host A is trying to communicate with Host C and sends a packet with the Host C's destination MAC address. The packet arrives at the switch, which looks at the destination MAC address. The

switch then searches that MAC address in its MAC address table. If the MAC address is found, the switch then forwards the packet only out the port connected to the frame's destination. Hosts connected to other ports will not receive the frame.

If the destination MAC address is not in the table, **the switch forwards the frame out all ports except the incoming port**. This is called an unknown unicast.

▼ CSMA/CD

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) helps hosts to decide when to send packets on a shared network segment and how to detect collisions if they occur. For example, in a hub network, two devices can send packets at the same time. This can cause a collision. CSMA/CD enables devices to "sense" the wire to ensure that no other device is currently transmitting packets. But, if two devices "sense" that the wire is clear and send packets at the same time, a collision can occur. If the collision occur, packets have to be resend after a random period of time.

Consider the following example:

In the topology above we have a hub network. Host A is trying to communicate with host B. Host A “senses” the wire and decides to send packets. But, in the same time, host C sends its packets to host D and the collision occurs. The sending devices (host A and host C) detect the collision and resend the packet after a random period of time.

NOTE: Since switches are now commonly used in networks instead of hubs, CSMA/CD is not really used anymore. Each port on a switch usually operate in a full duplex mode and there are no packet collisions in a full duplex mode.

▼ Collision & broadcast domain

Collision domain

A collision domain is, as the name implies, the part of a network where packet collisions can occur. A collision occurs when two devices send a packet at the same time on the shared network segment. The packets collide and both devices must send the packets again, which reduces network efficiency. **Collisions are often in a hub environment**, because each port on a hub is in the same collision domain. By contrast, **each port on a bridge, a switch or a router is in a separate collision domain**.

The following example illustrates collision domains:

We have 6 collision domains in the example above.

NOTE: Remember, each port on a hub is in the same collision domain. Each port on a bridge, a switch or router is in a separate collision domain.

Broadcast domain

A broadcast domain is the domain in which a broadcast is forwarded. A broadcast domain contains all devices that can reach each other at the data link layer (OSI layer 2) by using broadcast. All ports on a **hub or a switch** are by default in the **same broadcast domain**. All ports on a **router are in the different broadcast domains** and routers don't forward broadcasts from one broadcast domain to another.

The following example clarifies the concept:

In the picture above we have three broadcast domains, since all ports on a hub or a switch are in the same broadcast domain, and all ports on a router are in a different broadcast domain.

▼ How switches work?

Each network card has a unique identifier called a Media Access Control (MAC) address. This address is used in LANs for **communication between devices on**

the same network segment. Devices that want to communicate need to know each other MAC addresses before sending out packets.

Switches also use MAC addresses to make accurate forwarding or filtering decision. When a switch receives a frame, it associates the media access control (MAC) address of the sending device with the port on which it was received. The table that stores such associations is called a **MAC address table**. This table is stored in the volatile memory, so associations are erased after the switch is rebooted.

Switches usually perform these three functions in a LAN:

- **address learning** – switches learn MAC addresses by examining the source MAC address of each received frame.
- **forward/filter decisions** – switches decide whether to forward or filter a frame, based on the destination MAC address.
- **loop avoidance** – switches use Spanning Tree Protocol (STP) to prevent network loops while still permitting redundancy.

To better how a network switch works, take a look at the following example:

Let's say that host A wants to communicate with host B for the first time. Host A knows the IP address of host B, but since this is the first time the two hosts communicate, the hardware (MAC) addresses are not known. Host A uses the ARP process to find out the MAC address of host B. The switch forwards the ARP request out all ports except the port the host A is connected to. Host B

receives the ARP request and responds with its MAC address. Host B also learns the MAC address of host A (because host A sent its MAC address in the ARP request). Host C receives the ARP request, but doesn't respond since the IP address listed in the request is not its own.

As mentioned above, a switch learns which MAC addresses are associated with which port by examining the source MAC address of each received frame. Because host B responded with the ARP reply that included its MAC address, the switch knows the MAC address of host B and stores that address in its MAC address table. For host A, the switch knows its MAC address because of the ARP request that included it.

Now, when host A sends a packet to host B, the switch looks up in its MAC address table and forwards the frame only out the Fa0/2 port – the port on which host B is connected to. Other hosts on the network will not be involved in the communication:

NOTE: By default, MAC addresses stay in the switch's MAC address table for 5 minutes. So if host A and host B decide to communicate inside the next 5 minutes, a new ARP process will not be necessary.

You can display the MAC address table of the switch by using the *show mac-address-table* command:

```
Switch#show mac-address-table
          Mac Address Table
```

```
-----------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0003.e489.513e    DYNAMIC     Fa0/2
   1    00e0.8f13.6970    DYNAMIC     Fa0/1
```

The output is pretty much self-explanatory: all ports belong to VLAN 1 and MAC addresses associated with specific ports are listed. **DYNAMIC** means that the address were learned dynamically by using the source MAC address of the received frames.

> **VLANs (Virtual LANs)** are logical grouping of devices in the same broadcast domain. VLANs are usually configured on switches by placing some interfaces into one broadcast domain and some interfaces into another. Each VLAN acts as a subgroup of the switch ports in an Ethernet LAN.

▼ Spanning tree protocol

Spanning Tree Protocol (STP) is a network protocol designed to prevent **layer 2 loops**. It is standardised as IEEE 802.D protocol. STP blocks some ports on switches with redundant links to prevent broadcast storms and ensure loop-free topology. With STP in place, you can have redundant links between switches in order to provide redundancy.

To better understand the importance of STP and how STP prevents broadcast storms on a network with redundant links, consider the following example:

SW1 sends a broadcast frame to SW2 and SW3. Both switches receive the frame and forward the frame out every port, except the port the frame was received on. So SW2 forwards the frame to SW3. SW3 receives that frame, and forwards it to SW1. SW1 then again forwards the frame to SW2! The same thing also happens in the opposite direction. Without STP in place, these frames would loop forever. **STP prevents loops by placing one of the switch ports in blocking state.**

So, our toplogy above could look like this:

In the topology above, STP has placed one port on SW3 in the blocking state. That port will no longer process any frames except the STP messages. If SW3 receives a broadcast frame from SW1, it will not forward it out the port connected to SW2.

NOTE: STP enables layer 2 redundancy. In the example above, if the link between SW3 and SW1 fails, STP would converge and unblock the port on SW3.

▼ Layer 2 switching

Layer 2 switching (or **Data Link layer switching**) is the process of using devices' MAC addresses to decide where to forward frames. Switches and bridges are used for Layer 2 switching. They break up one large collision domain into multiple smaller ones.

In a typical LAN, all hosts are connected to one central device. In the past, the device was usually a hub. But hubs had many disadvantages, such as not being aware of traffic that passes through them, creating one large collision domain, etc. To overcome some of the problems with hubs, bridges were created. They were better than hubs because they created multiple collision domains, but they had limited number of ports. Finally, switches were created and are still widely used today. Switches have more ports than bridges, can inspect incoming traffic and make forwarding decisions accordingly. Also. each port on a switch is a separate collision domain, so no packet collisions should occur.

Layer 2 switches are faster than routers because they don't take up time looking at the Network layer header information. Instead, they look at the frame's hardware addresses to decide what to do with the frame – to forward, flood, or drop it. Here are other major advantages of Layer 2 switching:

- fast hardware-based bridging (using ASICs chips)
- wire speed
- low latency
- low cost

Here is an example of the typical LAN network – the switch serves as a central device that connects all devices together:

Differences between hubs and switches

To better understand the concept of frame switching based on the hardware address of a device, you need to understand how switches differ from hubs.

First, consider an example of a LAN in which all hosts connects to a hub:

As mentioned previously, hubs create only a single collision domain, so the chance for a collision to occur is high. The hub depicted above simply repeats the signal it

receives out all ports, except the one from which the signal was received, so no frame filtering takes place. Imagine if you had 20 hosts connected to a hub, a packet would be sent to 19 hosts, instead of just one! This can also cause security problems, because an attacker can capture all traffic on the network.

Now consider the way the switches work. We have the same topology as above, only this time we are using a switch instead of a hub:

Switches increase the number of collision domains. Each port is one collision domain, which means that the chances for collisions to occur are minimal. A switch learns which device is connected to which port and forwards a frame based on the destination MAC address included in the frame. This reduces traffic on the LAN and enhances security.

▼ Network router explained

A **router** is a network device that routes packets from one network to another. It is usually connected to two or more different networks. When a packet comes to a router port, the router reads the **address information (OSI layer 3 i.e. IP address) in the packet** to determine out which port the packet will be sent. For example, a router provides you with the internet access by connecting your LAN with the Internet.

Routers do not have a MAC address. A MAC address is only associated with a network interface component.

NOTE: A router is most commonly an OSI Layer 3 device, since its forwarding decision is based on the information of the OSI Layer 3 – the destination IP address. Routers divide broadcast domains, provide full duplex communication, and have traffic filtering capabilities.

The picture below shows a typical home router:

If two hosts from different networks want to communicate, they will need a router in order to exchange data. Consider the following example:

We have a network of three hosts and a router. Note that each computer is on a different network. Host A wants to communicate with Host B and sends the packet with the Host B's IP address (10.0.0.20) to the router. The router receives the packet, **compares the packet's destination IP address to the entries in its routing table and finds a match**. It then sends the packet out the interface associated with the network 10.0.0.0/24. Only Host B will receive and process the packet. In fact, Host C will not even be aware that the communication took place.

If the network containing the IP address is **not found**, then the router sends the packet on a default route, usually up the backbone hierarchy to the next router. Hopefully the next router will know where to send the packet. If it does not, again the packet is routed upwards until it reaches a **NSP backbone**. The routers connected to the NSP backbones hold the largest routing tables and here the packet will be routed to the correct backbone, where it will begin its journey 'downward' through smaller and smaller networks until it finds it's destination."

▼ Layer 3 switch and its working

A Layer 3 switch is a special network device that has the functionality of a router and a switch combined into one chassis. It works in our network by simply allowing connected devices that are on the same subnet or virtual LAN (VLAN) to exchange

information at lightning speed, just like a switch that operates in the data link layer of the OSI model, but it also has the IP routing intelligence of a router built into it.

It can inspect incoming packets in the network layer, support routing protocols, and even make routing decisions based on the source and destination IP addresses. With **both its Layer 2 and Layer 3 capabilities, this device is popularly known also as a Multilayer Switch**. Just be mindful that Layer 3 switches do not have WAN ports which should be considered while designing your network because it is designed specifically for use on intranets

Wide Area Network port is the **socket on a network device that is wired to an external network**, which is typically the Internet.

How Layer 3 Switches Functions in the Network?

Layer 2 switch dynamically routes traffic between its physical interfaces according to the MAC addresses of the connected devices, wherein Layer 3 switches use this feature to manage traffic in a LAN. A **Layer 2 switch functions well in low to medium traffic in its VLANs, but these switches have their limitations once traffic increases.**

The Layer 3 switch was conceived to augment this limitation by developing equipment that has routing capabilities within the same chassis. The hardware is where the main difference lies. Layer 3 switches have a mix of traditional switches and routers, except for the fact that the router's software logic is replaced by integrated circuit hardware to improve its performance further.

Layer 3 switches can perform on the OSI model's Layer 2 and Layer 3. The Layer 3 switching functionality can take either of two forms:

- **Cut-through switches** – will only look into the first packet of a series of packets to determine its logical Layer 3 destination IP address and then shift

the remainder of the packets in the series using the MAC address leading to higher data throughput rates.

- **Packet-by-Packet Layer 3 (PPL3) switches** – will look into every packet to determine its logical Layer 3 destination IP address. A PPL3 switch basically functions as a high-speed router with the routing functionality built into its hardware instead of software. Similar to routers, aside from forwarding packets to their destination, PPL3 switches perform other functions that a standard router accomplishes, such as using the packet's checksum to verify its integrity, updating the packet's Time to Live (TTL) information after each hop, and processing any optional information in the packet's header.

In addition to performing Layer 3 switching functions and routing functions, these switches perform the Layer 2 switches functions, such as bridging functions, at each switch interface. You can group switching interfaces in various ways to allocate bandwidth and contain broadcasts, which makes Layer 3 switches a powerful, scalable technology for building high-speed Ethernet backbone networks.

L3 Switch Benefits

Layer 3 switches were developed to provide the network with the following advantages:

- Better fault isolation and traffic segregation
- Simplify security management
- Reduce broadcast traffic volume
- Easier VLAN configuration process
- Support Inter-VLAN routing
- Separate routing tables
- Reduce effort and time in troubleshooting
- Support flow accounting and high-speed scalability
- Lower network latency

> A VLAN (virtual LAN) is a subnetwork which can group together collections of devices on separate physical local area networks (LANs). Devices in VLANs are logically connected and not physically.

▼ Network bridge explained

A **network bridge** is a device that divides a network into segments. Each segment represent a separate collision domain, so the **number of collisions on the network is reduced**. Also, because each collision domain has its own separate bandwidth, a bridge also improves the overall network performance.

NOTE: Unlike hubs, bridges allow multiple devices to send at the same time. This is why there are considered to be predecessors of network switches.

A bridge works at the Data link layer (**Layer 2**) of the OSI model, just like a switch does. It inspects incoming traffic and decide whether to forward it or filter it. Each incoming Ethernet frame is inspected for destination MAC address. If the bridge determines that the destination host is on another segment of the network, it forwards the frame to that segment.

Consider the following network:

In the example above we have a network of four computers. The network is divided into segments by a bridge. Each segment is a separate collision domain with its own bandwidth. Let's say that Host A wants to communicate with Host C. Host A will send the frame with the Host C's destination MAC address to the

bridge. The bridge will inspect the frame and forward it to the segment of the network Host C is on.

Network bridges offered substantial improvements over network hubs, but are not widely used anymore in modern LANs – switches are commonly used instead. Here is why:

- most bridges have only 2 or 4 ports. A switch can have tens or even hundreds of ports
- bridges are software based, while switches are hardware-based and use chips (ASICs) when making forwarding decisions, which makes them much faster than bridges
- switches can have multiple spanning-tree instances, bridges can have only one
- switches can have multiple broadcast domains (one per VLAN)

▼ Gateway

A gateway is **a network node used in telecommunications that connects two networks with different transmission protocols together**. Gateways serve as an entry and exit point for a network as all data must pass through or communicate with the gateway prior to being routed.

▼ Life of a packet

▼ Domain Name System and its working

Computers communicate with each other by using an IP address (IPv4 or IPv6), but how come we are using hostnames like www.google.com to access the web server of Google where their website is hosted? By using DNS, we can be able to access websites with easy-to-remember names.

Imagine always remembering IP addresses to access specific websites. That is not a convenient way by typing only a hostname on the web browser like www.google.com instead of typing 142.250.181.46 (Google web server's public IP address), which will simplify the process of accessing the website.

What is a Domain Name System (DNS)?

DNS is a decentralized and hierarchical naming convention system for computers or other resources that are connected to the Internet or to the Local Area Network (LAN). It operates on the **application layer** of OSI model. It uses a hostname to identify other computers. DNS solves the issue of remembering all the IP addresses by converting (resolving) them into a hostname that the website administrator can customize. The primary purpose of DNS is to resolve the hostnames to the IP addresses. The Uniform Resource Locator (URL) or a domain name of the website is a lot easier to remember compared to the IP address of the website.

Before the DNS is being implemented, the computer can use a domain name by using a host file. The **host file** contains the hostname and maps it to a specific IP address. Whenever the computer wants to visit a website on the internet, it will check first on the host file and map it to the IP address of the website. What if the hostname of the website or its IP address is not registered on the host file? The computer will not be able to connect to the website.

Frequently updating the host file is not a convenient and efficient way as the internet is continuously growing. To solve the issue, a **DNS Server (Name Server)** was created. The DNS servers are being the root servers for its domain and contain all the DNS records for the specific domain like TLD. **Top-Level Domain (TLD)** is a domain that contains a root (.) and ends name like .net, .com, or .org. On the other hand, the Fully Qualified Domain Name (FQDN) contains a hostname, domain name, and TLD. When accessing www.google.com, "www" is the hostname, "google" is the domain name, and ".com" is the TLD.

NOTE: A **hostname** represents a network used to deliver a user to a specific address, while a **domain name** is a site that the user is accessing.

Domain Name System Operation

While internal DNS and caching is common in most enterprises, let's look at the simplest form of DNS. When the request for an IP is initiated by a user by typing a URL in a browser, the request first goes to a recursive resolver -- if it's not already in the browser's or operating system's caches -- which is a server usually managed either by the user's ISP, a managed DNS provider or the enterprise itself. The **resolver knows** which DNS servers it will need to ask to unlock the IP address, and will generally forward the request to a root server that knows all of the information **about top-level domains** (.com, .net, .org, etc.). Because these

are spread out globally, the resolver will generally access the root server closest to the geographic start of the request -- that is, closest to the user accessing the web.

The next step in the hierarchy takes the request to the **top-level domain (TLD) server**. This server gets more specific, **recognizing the secondary domain** (the "google" in "www.google.com"), and then sends the request off to that domain's **authoritative name server, which ultimately holds the keys to the IP** address that allows the user to visit the website.

Despite all of the stop-points in the DNS hierarchy, this whole journey should happen in the blink of an eye (read: milliseconds). Because it's such an efficient process, the average Internet user outside of the networking realm takes DNS for granted.

> **User → DNS resolver cache → 13 Root server → TLD Name server → Thousands of Authoritative Name Server**

What if the DNS does not have the record for the specific hostname?

The DNS server will request and **exchange information with other DNS servers** located somewhere on the Internet and will respond to the customer with the IP address of the hostname. The DNS requests are sent using the **UDP port 53**. However, it can failover and use TCP.

Why TCP is not used in DNS?

Basically, to establish connection every time 3 packets need to be send this increases the traffic while that is not in the case of udp as connection is not required. In case, it misses any packet it would request again.

https://www.youtube.com/watch?v=vuyQ1PW6AwY

▼ ARP (Address Resolution Protocol) explained

ARP (Address Resolution Protocol) is a network protocol used to find out the hardware (MAC) address of a device from an IP address. It is used when a device wants to communicate with some other device on a local network (for example on an Ethernet network that requires physical addresses to be known before sending packets). The sending device uses ARP to translate IP addresses to MAC addresses. The **device sends an ARP request message containing the IP address of the receiving device**. All devices on a local network segment see the message, but only the device that has that IP address responds with the ARP reply message containing its MAC address. The sending device now has enough information to send the packet to the receiving device.

ARP request packets are sent to the broadcast addresses (FF:FF:FF:FF:FF:FF for the Ethernet broadcasts and 255.255.255.255 for the IP broadcast).

Here is the explanation otf the ARP process:

Let's say that Host A wants to communicate with host B. Host A knows the IP address of host B, but it doesn't know the host B's MAC address. In order to find out the MAC address of host B, host A sends an ARP request, listing the host B's IP address as the destination IP address and the MAC address of FF:FF:FF:FF:FF:FF (Ethernet broadcast). Switch will forward the frame out all interfaces (except the incoming interface). Each device on the segment will receive the packet, but because the destination IP address is host B's IP address, **only host B will reply with the ARP reply packet, listing its MAC address**. Host A now has enough information to send the traffic to host B.

All operating systems maintain ARP caches that are checked before sending an ARP request message. Each time a host needs to send a packet to another host on the LAN, it first checks its ARP cache for the correct IP address and matching MAC address. The addresses will stay in the cache for a couple of minutes. You can display ARP entries in Windows by using the *arp -a* command:

ARP is replaced by ip neighbour command.

Linux networking: arp versus ip neighbour

Address resolution protocol, aka ARP, is a communication protocol used in IPv4 networks. It was developed in 1982 and has been a staple in most networks since that time. The same

https://www.redhat.com/sysadmin/arp-versus-ip

▼ ARP table on router

Just like regular hosts, if a Cisco router wants to exchange frames with a host in the same subnet, it needs to know its MAC address. The IP-to-MAC address mapping are kept in the router's ARP table. Consider the following example:

R1 has two connected subnets – 10.0.0.0/24 and 172.16.0.0./16. Before exchanging frames with either host, R1 will need to know their MAC addresses. Here is the output of the R1's ARP table:

```
R1#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.0.0.1                -   0060.5C32.7E01  ARPA   GigabitEthernet0/0
Internet  10.0.0.10               6   000C.85CA.AD73  ARPA   GigabitEthernet0/0
```

```
Internet  172.16.0.1              -   0060.5C32.7E02  ARPA   GigabitEthernet0/1
Internet  172.16.0.2             10  0001.63DB.1802  ARPA   GigabitEthernet0/1
```

The ARP table contains two entries for R1's own two interfaces with the IP address of 10.0.0.1 and 172.16.0.1. The "–" in the age column indicates that the entry will never be timed out.

The ARP table also lists the MAC addresses of the two connected hosts. Consider the entry for Host A:

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.0.0.10               6   000C.85CA.AD73  ARPA   GigabitEthernet0/0
```

Here is a brief description of each field:

- **Protocol** – the protocol type, almost always Internet
- **Address** – the IP address associated with the MAC address, in our case the IP address of Host A
- **Age** – by default, an entry will be removed from the ARP table if it wasn't used in 240 minutes. 6 in this column means that the entry was last used 6 minutes ago. Each time an entry is used, the age will be reset back to zero.
- **Hardware** – the MAC address of the host with the corresponding IP address.
- **Type** – the type of hardware address. For Ethernet, this value will always be ARPA.
- **Interface** – the interface on R1 on which the corresponding host is connected.

Here are the steps R1 needs to take before forwarding frames to Host A:

1. R1 wants to communicate with Host A. R1 checks its routing table. The subnet on which Host A resides is a directly connected subnet.
2. R1 checks its ARP table to find out whether the Host A's MAC address is known. If it is not, R1 will send an ARP request to the broadcast MAC address of FF:FF:FF:FF:FF:FF.
3. Host A receives the frame and sends its MAC address to R1 (ARP reply). The host also updates its own ARP table with the MAC address of the Gigabit0/0 interface on R1.

4. R1 receives the reply and updates the ARP table with the MAC address of Host A.
5. Since both hosts now know each other MAC addresses, the communication can occur.

▼ Network troubleshooting

▼ ICMP (Internet Control Message Protocol)

ICMP is a **network layer** protocol that reports errors and provides information related to IP packet processing. ICMP is used by network devices to send error messages indicating, for example, that a requested service is not available or that a host isn't reachable.

ICMP is commonly used by network tools such as ping or traceroute. Consider the following example that illustrates how ping can be used to test the reachability of a host:

Host A wants to test whether it can reach **Server** over the network. Host A will start the ping utility that will send **ICMP Echo Request** packets to Server. If Server is reachable, it will respond with **ICMP Echo Reply** packets. If Host A receives no response from Server, there might be a problem on the network.

NOTE: ICMP messages are encapsulated in IP datagrams, which means that they don't use higher level protocols (such as TCP or UDP) for transmission.

One other common ICMP message is the **Destination unreachable** message. Here is an example:

Host A sends a packet to Host B. Because the Host B is down, the router will send an **ICMP Destination host unreachable** message to Host A, informing it that the destination host is unreachable, e.g.:

```
C:\>ping 192.168.8.11

Pinging 192.168.8.11 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.8.11:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

▼ Ping explained

ping is perhaps the most commonly used tool to troubleshoot a network. **Ping (Packet Internet Groper)** is included with most operating systems. It is invoked using a ping command and uses **ICMP (Internet Control Message Protocol)** to reports errors and provides information related to IP packet processing. Ping works by sending an ICMP echo request message to the specified IP address. If the computer with the destination IP address is reachable, it responds with an ICMP echo reply message.

A ping command usually outputs some other information about a network performance, e.g. a round-trip time, a time to send an ICMP request packetand receive an ICMP reply packet.

Here is an output of the ping command from a Windows PC:

```
Command Prompt

C:\Users\user>ping 10.10.100.1

Pinging 10.10.100.1 with 32 bytes of data:
Reply from 10.10.100.1: bytes=32 time<1ms TTL=128
Reply from 10.10.100.1: bytes=32 time<1ms TTL=128
Reply from 10.10.100.1: bytes=32 time<1ms TTL=128
Reply from 10.10.100.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.100.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\user>_
```

In the example above we have pinged the ip address **10.10.100.1**. By default, ping on Windows sends four ICMP request packets. As you can see from the output above, the host with the IP address of 10.10.100.1 is reachable and has replied with four ICMP reply packets. You can also see that the remote host has replied within 1 ms (time<1ms), which indicates that the network is not congested.

▼ Ping commands

- ping www.google.com

  It will ping the google search engine. Whose public id currently is 142.250.192.36.

```
khush@khush-HP-ProBook-430-G3:~$ ping www.google.com
PING www.google.com (142.250.192.36) 56(84) bytes of data.
64 bytes from bom12s15-in-f4.1e100.net (142.250.192.36): icmp_seq=1 ttl=60 time=10.8 ms
64 bytes from bom12s15-in-f4.1e100.net (142.250.192.36): icmp_seq=2 ttl=60 time=10.6 ms
64 bytes from bom12s15-in-f4.1e100.net (142.250.192.36): icmp_seq=3 ttl=60 time=11.5 ms
64 bytes from bom12s15-in-f4.1e100.net (142.250.192.36): icmp_seq=4 ttl=60 time=11.3 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 10.587/11.061/11.536/0.382 ms
```

- ping 8.8.8.8

  It will ping the google DNS server.

1. `from`: The destination and its IP address. Note that the IP address may be different for a website depending on your geographical location.

2. `icmp_seq=1`: The sequence number of each ICMP packet. Increases by one for every subsequent echo request.
3. `ttl=60`: The Time to Live value from 1 to 255. It represents the number of network hops a packet can take before a router discards it.

   What is TTL shown in ping? First some background on why TTL was invented. Every bit of data on IP networks is sent using packets, and there are almost always one or more routers in between two computers speaking to each other on the internet. It is the router's job to move these packets closer to their destination. However, sometimes things can go wrong, and routers can end up forwarding packets back and forth to one another forever. This is called a routing loop, and, there need to be protections in place to prevent these packets from being forever passed back and forth in the network.

   Enter "TTL", or "time to live". TTL is a bit of data kept with every packet, indicating how many more routers, or hops, it is allowed to pass between before expiring. For every router a packet passes through, this TTL value will be decreased by 1. When the value reaches zero, the router handling that packet will drop the packet and will send a warning message back to the sender, letting them know the TTL expired.
4. `time=10.8 ms`: The time it took a packet to reach the destination and come back to the source. Expressed in milliseconds.

OPTIONS: -4, -c, -w, -i, -s, -f, -a, -v

Syntax of the ping command

The basic syntax of the ping command is as follows:

$ ping host-name/IP

```
kbuzdar@kbuzdar: ~
kbuzdar@kbuzdar:~$ ping linuxhint.com
PING linuxhint.com (64.91.238.144) 56(84) bytes of data.
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=1 ttl=49 time=307 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=2 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=3 ttl=49 time=310 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=4 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=5 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=6 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=7 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=8 ttl=49 time=312 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=9 ttl=49 time=305 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=10 ttl=49 time=309 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=11 ttl=49 time=305 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=12 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=13 ttl=49 time=311 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=14 ttl=49 time=305 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=15 ttl=49 time=305 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=16 ttl=49 time=306 ms
```

Use Ctrl + C from your keyboard to terminate the process.

```
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=298 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=299 ttl=49 time=306 ms
^C
--- linuxhint.com ping statistics ---
299 packets transmitted, 296 received, 1.00334% packet loss, time 298220ms
rtt min/avg/max/mdev = 304.165/309.650/524.322/18.924 ms
kbuzdar@kbuzdar:~$
```

Check local network using the ping command

Sometimes, you encounter problems in reaching the desired website. So, in this situation, you can ping your localhost or computer system to check the proper internet connection. The following commands are useful to check your local internet connection:

$ ping localhost

Or

$ ping your-IP-address

In the following output, you have pinged your local machine using two different ways. In the first, you use the localhost that means ‘ping to my computer’.

```
kbuzdar@kbuzdar: ~
kbuzdar@kbuzdar:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.070 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.074 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.069 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.045 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.073 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.069 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.075 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.069 ms
64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=0.075 ms
^C
--- localhost ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9196ms
rtt min/avg/max/mdev = 0.019/0.063/0.075/0.017 ms
```

In the below output, using IP address ping the local computer system.

```
kbuzdar@kbuzdar: ~
kbuzdar@kbuzdar:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.070 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.074 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.074 ms
^C
```

Using ping specify internet protocol

The ping command is also used to send request IPv4 or IPv6 as follows:

$ ping -6 hostname/IPv6

or

$ ping -4 hostname/IPv4

```
kbuzdar@kbuzdar:~$ ping -6 127.0.0.1
ping: 127.0.0.1: Address family for hostname not supported
kbuzdar@kbuzdar:~$ ping -4 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.071 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.083 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.035 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.048 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.137 ms
^C
--- 127.0.0.1 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7158ms
rtt min/avg/max/mdev = 0.020/0.063/0.137/0.034 ms
kbuzdar@kbuzdar:~$
```

Change ping packets time interval

The **default time interval between the requesting packet is 1 second in the Linux**. But, according to your requirements, you can increase or decrease this time interval.

If you want to increase this time interval then you need to set above 1 as follows:

$ ping -i 3 host-name/IP

If you want to decrease the interval then you will set the time less than 1 as follows:

$ ping -i 0.5 linuxhint.com

```
kbuzdar@kbuzdar: ~
kbuzdar@kbuzdar:~$ ping -i 0.5 linuxhint.com
PING linuxhint.com (64.91.238.144) 56(84) bytes of data.
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=1 ttl=49 time=308 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=2 ttl=49 time=320 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=3 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=4 ttl=49 time=305 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=5 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=6 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=7 ttl=49 time=315 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=8 ttl=49 time=331 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=9 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=10 ttl=49 time=305 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=11 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=12 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=13 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=14 ttl=49 time=307 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=15 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=16 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=17 ttl=49 time=305 ms
^C
--- linuxhint.com ping statistics ---
18 packets transmitted, 17 received, 5.55556% packet loss, time 8526ms
rtt min/avg/max/mdev = 305.054/308.791/330.881/6.717 ms
kbuzdar@kbuzdar: $
```

Change ping packet size

Using the ping command, you can change the size of ping packets. The default size of the packet is set to 56 (84) bytes. The parentheses number represents that on sending ping bytes 28 packet header bytes will also include in the packet size.

> Where, 20 (IP header size) + 8 (ICMP header size) = 28 (ethernet frame)

You can change this size using the following command:

$ ping -s packetsize hostname/IP

The above command is used to test network performance.

For example, we have changed the ping packet size to 150 as follows:

$ ping -s 150 64.9.238.144

```
kbuzdar@kbuzdar:~$ ping -s 150 64.91.238.144
PING 64.91.238.144 (64.91.238.144) 150(178) bytes of data.
158 bytes from 64.91.238.144: icmp_seq=1 ttl=49 time=307 ms
158 bytes from 64.91.238.144: icmp_seq=2 ttl=49 time=316 ms
158 bytes from 64.91.238.144: icmp_seq=3 ttl=49 time=316 ms
158 bytes from 64.91.238.144: icmp_seq=4 ttl=49 time=306 ms
158 bytes from 64.91.238.144: icmp_seq=5 ttl=49 time=306 ms
158 bytes from 64.91.238.144: icmp_seq=6 ttl=49 time=340 ms
158 bytes from 64.91.238.144: icmp_seq=7 ttl=49 time=349 ms
^C
--- 64.91.238.144 ping statistics ---
8 packets transmitted, 7 received, 12.5% packet loss, time 7011ms
rtt min/avg/max/mdev = 306.083/320.139/349.363/16.136 ms
kbuzdar@kbuzdar:~$
```

Set limit to desire number of ping packets

If you want to set limits that ping request automatically stopped after the specified number of ping packets then you can use the following command:

$ ping –c set-limit host-name/IP

For example, set the ping request to 7 which means after sending 7 ping packets, the ping request will automatically terminate. You can see the output in the following image:

$ ping –c 7 127.0.0.1

```
kbuzdar@kbuzdar:~$ ping -c 7 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.073 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.076 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.069 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.075 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.072 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.072 ms

--- 127.0.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6135ms
rtt min/avg/max/mdev = 0.019/0.065/0.076/0.018 ms
kbuzdar@kbuzdar:~$
```

Using ping command flood a network

Under the heavy load, you can test your network performance using the ping flood. The ping flood must be executed under the root user. The ‘-f’ is used with ping command as follows:

$ **sudo** ping -f host-name/IP

```
kbuzdar@kbuzdar: ~
kbuzdar@kbuzdar:~$ sudo ping -f linuxhint.com
[sudo] password for kbuzdar:
PING linuxhint.com (64.91.238.144) 56(84) bytes of data.
....................^C
--- linuxhint.com ping statistics ---
1809 packets transmitted, 1790 received, 1.0503% packet loss, time 28577ms
rtt min/avg/max/mdev = 303.046/306.440/371.076/3.965 ms, pipe 23, ipg/ewma 15.80
5/306.463 ms
kbuzdar@kbuzdar:~$
```

Using this command, **large packets send to the network for testing performance**.

The dots represent the sending packet and it prints backspace for each response.

Set timeout for a ping request

Using the ping command, you can also set a timeout limit for ping requests. It means that after the specified time the ping will stop displaying the results on the terminal. Use the following command to check results:

$ ping -w time-in-seconds host-name/IP

Here, after 6 seconds ping automatically stops to print results.

$ ping -w 6 linuxhint.com

```
kbuzdar@kbuzdar:~$ ping -w 6 linuxhint.com
PING linuxhint.com (64.91.238.144) 56(84) bytes of data.
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=1 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=2 ttl=49 time=305 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=3 ttl=49 time=307 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=4 ttl=49 time=305 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=5 ttl=49 time=306 ms
64 bytes from linuxhint.com (64.91.238.144): icmp_seq=6 ttl=49 time=306 ms

--- linuxhint.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5003ms
rtt min/avg/max/mdev = 304.958/305.764/307.297/0.745 ms
```

Ping with an audible response

Using the following ping command, you can set a beep play that will check that the host available or not:

$ ping -a host-name/IP

Check ping version

You can check the ping utility version using the following command:

$ ping –V

```
kbuzdar@kbuzdar:~$ ping -V
ping from iputils s20190709
kbuzdar@kbuzdar:~$
```

▼ Extended Ping command

The *ping* command in Cisco IOS (and other operating systems) is used to test the accessibility of devices on a TCP/IP network. Cisco devices also support the *extended ping* command that allows you to perform a more advanced check of the host reachability and network connectivity. With this command, you can define the source IP address as any IP address on the router, number and size of ping packets, different timeout interval, etc.

The *extended ping* command is invoked from the privileged exec mode by typing *ping* and pressing Enter. The following arguments can be modified:

- **Protocol [ip]** – specify the protocol, such as appletalk, clns, ip, novell, apollo, vines, decnet, or xns. The default is *ip*.
- **Target IP address** – specify the IP address or the hostname of the host to ping.
- **Repeat count** – specify the number of ping packets that will be sent to the destination address. 5 by default.
- **Datagram size** – specify the size of the ping packet (in bytes). The default is 100 bytes.
- **Timeout in seconds** – specify the timeout interval. The default is 2 seconds. The echo reply needs to be received before the timeout expires in order for ping to be successful.

- **Extended commands** – specify whether or not a series of additional commands will appear. The default is *no*. If you type *yes* additional arguments will be shown.
- **Source address or interface** – specify the interface or the IP address of the router to use as the source address for the ping packets.
- **Type of service** – specifies the Type of Service (ToS). This is the Internet service's quality selection. The default is 0.
- **Set DF bit in IP header?** – specify whether or not the Don't Fragment (DF) bit will be set on the ping packet. If yes is entered, the Don't Fragment option does not allow the packet to be fragmented. The default is no.
- **Validate reply data?** – specify whether or not to validate the reply data. The default is no.
- **Data pattern** – specify the data pattern. Data patterns are used to troubleshoot framing errors and clocking problems on serial lines. The default is *[0xABCD]*.
- **Loose, Strict, Record, Timestamp, Verbose** – specify the IP header options.
- **Sweep range of sizes** – specify the sizes of the ping echo packets that are sent. This parameter is used to determine the minimum sizes of the MTUs configured on the nodes along the path to the destination address. The default is no.

The extended ping command is most often used to change the source IP address of the ping echo packets. Consider the following example:

By default, routers choose the IP address of the outgoing interface as the source IP address for ping echo packets. This means that R2 will use the IP address of the Fa0/1 interface (10.0.0.1) as the source IP address for the ping packets sent to 10.0.0.2 (R1). Let's try to ping R1 using the standard ping command:

```
R2#ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms
```

As you can see from the output, the ICMP replies were received. However, I can run the *extended ping* command to change the source IP address to the IP address of the R2 Fa0/0 interface (192.168.5.1). This is done to ensure that R1 knows about the 192.168.5.1 network (in other words, that it knows where to send packets destined for the 192.168.5.0/24 network, which could indicate routing problems).

```
R2#ping
Protocol [ip]:
Target IP address: 10.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: yes
Source address or interface: 192.168.5.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.5.1
.....
Success rate is 0 percent (0/5)
```

In the output above you can see that no echo replies were received after I've changed the source IP of the ping packets. This means that R1 doesn't know how to reach the 192.168.5.0/24 network.

▼ Traceroute

Traceroute is a command-line interface based tool used to identify the path used by a packet to reach its target. This tool also uses ICMP messages, but unlike ping, it **identifies every router in a path taken by the packets**. Traceroute is useful when troubleshooting network problems because it can help identify where exactly the problem is. You can figure out which router in the path to an unreachable target should be examined more closely as the probable cause of the network's failure.

Traceroute sends a series of ICMP echo request packets to a destination. First series of messages has a Time to Live (TTL) parameter set to 1, which means that the first router in a path will discard the packet and send an ICMP Time Exceeded message. TTL is then increased by one until the destination host is reached and an ICMP echo reply message is received. Originating host can then use received ICMP messages to identify all routers in a path.

Here is an example of using the **traceroute** command:

```
khush@khush-HP-ProBook-430-G3:~$ traceroute google.com
traceroute to google.com (142.250.67.174), 30 hops max, 60 byte packets
 1  dlinkrouter (192.168.0.1)  2.246 ms  5.416 ms  5.396 ms
 2  10.112.0.1 (10.112.0.1)  13.814 ms  14.221 ms  15.004 ms
 3  103.241.47.41 (103.241.47.41)  14.929 ms  15.815 ms  15.753 ms
 4  142.250.47.236 (142.250.47.236)  17.767 ms  18.923 ms  18.955 ms
 5  * * *
 6  142.250.210.182 (142.250.210.182)  20.981 ms 142.250.60.134 (142.250.60.134)
13.000 ms 108.170.234.208 (108.170.234.208)  11.247 ms
 7  108.170.248.210 (108.170.248.210)  11.924 ms 142.250.227.73 (142.250.227.73)
13.525 ms 108.170.248.218 (108.170.248.218)  16.313 ms
 8  108.170.248.177 (108.170.248.177)  16.258 ms 108.170.248.161 (108.170.248.161)
  16.200 ms bom12s07-in-f14.1e100.net (142.250.67.174)  16.144 ms
```

OUTPUT: Hop-number Name ip-address rtt1 rtt2 rtt3

Hop number : The specific hop number in the path from the sender to the destination.

Round Trip Time (RTT): The time it takes for a packet to get to a hop and back, displayed in milliseconds (ms). By default, **traceroute** sends three packets to each hop, so the output lists three roundtrip times per hop. RTT is sometimes also referred to as latency. An important factor that may impact RTT is the physical distance between hops.

In the output above you can see that the **traceroute** command has listed the IP addresses of all of the routers in the path.

Traceroute on Unix-like operating systems

Traceroute command on Unix works slighty different than the Windows version. It uses UDP packets with a large destination port number (33434 to 33534) that is unlikely to be used by any application at the destination host. Like the Windows version of the command, traceroute on Unix uses TTL to get the IP addresses of the intermediary routers. When a destination host is reached, it replies with an ICMP port unreachable message.

▼ traceroute command

The *traceroute* command on a Cisco device can be used to identify the path used by a packet to reach its target. It identifies all the routers in the path from the source host to destination host and it can be useful when troubleshooting network problems. Using this command you can figure out which router in the path to an unreachable target should be examined more closely as the probable cause of the network's failure.

Consider the following example network:

In the picture above you can see that we have a network of four routers. The network is working correctly. Consider what happens when we issue the *traceroute* command on R1 to the IP address of the router's R4 Gi0/0 interface (172.16.0.2):

```
R1#traceroute 172.16.0.2
Type escape sequence to abort.
Tracing the route to 172.16.0.2
1 10.0.0.1 0 msec 0 msec 0 msec
2 192.168.5.2 0 msec 0 msec 0 msec
3 172.16.0.2 0 msec 0 msec 0 msec
```

You can see that the traceroute command has listed the IP addresses of all of the routers in the path to the destination. Now let's assume that the router R3 goes down. Consider the command output now:

```
R1#traceroute 172.16.0.2
Type escape sequence to abort.
Tracing the route to 172.16.0.2
1 10.0.0.1 0 msec 0 msec 0 msec
2 10.0.0.1 !H * !H
3 * *
```

Notice how there is no response from R3 (192.168.5.2). Using this information, we can assume that there is a problem with R3 and investigate the matter. The *!H* in the output indicates that the host is unreachable. Other character that can appear in the *traceroute* command output are:

- number of miliseconds – t**he round-trip time in milliseconds.**
- – the probe has timed out.
- A – administratively prohibited (for example, with an access-list).
- Q – source quench (the destination is too busy).
- I – user interrupted test.
- U – port is unreachable.
- N – the network is unreachable.
- P – the protocol is unreachable.
- T – timeout.
- ? – unknown packet type.

▼ SNMP

https://www.youtube.com/watch?v=Lq7j-QipNrI

Network Monitoring Software by ManageEngine OpManager

Simple Network Management Protocol (SNMP) is an application-layer protocol defined by the Internet Architecture Board (IAB) in RFC1157 for exchanging management information between network

https://www.manageengine.com/network-monitoring/what-is-snmp.html

▼ SSH

```
ssh usename@ip_address
Password: ******

Done. Now can use their terminal using commands
```

Telnet is the standard TCP/IP protocol for virtual terminal service, while SSH or Secure Shell is a program to log into another computer over a network to execute commands in a remote machine. **Telnet transfers the data in** plain text while in SSH data is sent in encrypted format via a secure channel.

https://www.youtube.com/watch?v=ORcvSkgdA58

9 - Cryptography Basics - SSH Protocol Explained

In this video you'll learn about the details of SSH protocol.

https://www.youtube.com/watch?v=0Sffl7YO0aY

What is SSH?

SSH, or Secure Shell, is a network protocol that allows one computer to securely connect to another computer over an unsecured network, like the internet, by having a shared

https://levelup.gitconnected.com/what-is-ssh-103f89e3e4b8

Introduction to SSH(Secure Shell) Keys - GeeksforGeeks

SSH(Secure Shell) is access credential that is used in the SSH Protocol. In other words, it is a cryptographic network protocol that is used for transferring encrypted data over network. It allows

https://www.geeksforgeeks.org/introduction-to-sshsecure-shell-keys/

▼ Basics & tcpdump

- bandwidth delay product = mbps * channel delay

  how many bits can be there on channel.

- If we want to send mail to a group it goes to the gmail server now it is multicast enabled uses igmp now it will create copies of msg and send all. Now i will not have to create many copies and do bombarding of packets from my system.

  smtp simple mail transfer protocol for sending message. pop post office protocol & imap is used when we open gmail and read mail.

- IP protocol is connection less. It is called best effort delivery protocol. To maintain a connection you need to store some data at the router. So if ip is kept connection oriented then. Huge amount of data might get at each router,

- data link layer is made up of 2 layer: logic link control (error control and flow control) and media access control (tightly coupled with physical layer)

  wired network: csma/cd (collision detection)

  wireless:csma/ca (collision avoidance)

- **Changing ip address:** sudo ifconfig enp0s8 192.168.1.12 netmask 255.255.255.0 up

- iwconfig is ifconfig for wireless

  iwconfig wlan0 down/up : connect or disconnect connection

- Find name of device using ip address: nslookup 192.168.0.133

- tcpdump it is used for packet capturing/analysis

  tcpdump -D : show all interfaces to capture traffic

  tcpdump -i wlan0: i is for interface

tcpdump -i wlan0 -c 5: capture 5 packets

tcpdump -i wlan0 -c 5 icmp: capture icmp packets || in other terminal ping facebook.com

Beacuse ping works on icmp

```
khush@khush-HP-ProBook-430-G3:~$ sudo tcpdump -i wlp2s0 -c 5 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlp2s0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:42:21.496329 IP khush-HP-ProBook-430-G3 > edge-star-mini-shv-02-bom1.facebook.com: ICMP echo request, id 20, seq 1, length 64
20:42:21.512238 IP edge-star-mini-shv-02-bom1.facebook.com > khush-HP-ProBook-430-G3: ICMP echo reply, id 20, seq 1, length 64
20:42:22.498245 IP khush-HP-ProBook-430-G3 > edge-star-mini-shv-02-bom1.facebook.com: ICMP echo request, id 20, seq 2, length 64
20:42:22.514031 IP edge-star-mini-shv-02-bom1.facebook.com > khush-HP-ProBook-430-G3: ICMP echo reply, id 20, seq 2, length 64
20:42:23.499205 IP khush-HP-ProBook-430-G3 > edge-star-mini-shv-02-bom1.facebook.com: ICMP echo request, id 20, seq 3, length 64
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

- tcpdump -i wlan0 port 80: capture htttp packets || in browser search for any http website like http://info.cern.ch || or download a webpage usin wget http://tutorials.jenkov.com/

```
khush@khush-HP-ProBook-430-G3:~$ sudo tcpdump -i wlp2s0 -c 5 port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlp2s0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:46:43.999598 IP khush-HP-ProBook-430-G3.45522 > webafs706.cern.ch.http: Flags [S], seq 1085870567, win 64240, options [mss 1460,sackOK,TS va
l 2754906132 ecr 0,nop,wscale 7], length 0
20:46:44.000842 IP khush-HP-ProBook-430-G3.45520 > webafs706.cern.ch.http: Flags [P.], seq 153853761:153854340, ack 2128576488, win 502, option
s [nop,nop,TS val 2754906134 ecr 3365677384], length 579: HTTP: GET / HTTP/1.1
20:46:44.152856 IP webafs706.cern.ch.http > khush-HP-ProBook-430-G3.45522: Flags [S.], seq 768363677, ack 1085870568, win 28960, options [mss 1
452,sackOK,TS val 3365686336 ecr 2754906132,nop,wscale 7], length 0
20:46:44.152953 IP khush-HP-ProBook-430-G3.45522 > webafs706.cern.ch.http: Flags [.], ack 1, win 502, options [nop,nop,TS val 2754906286 ecr 33
65686336], length 0
20:46:44.155884 IP webafs706.cern.ch.http > khush-HP-ProBook-430-G3.45520: Flags [.], ack 579, win 236, options [nop,nop,TS val 3365686335 ecr
2754906134], length 0
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

- tcpdump -i wlan0 -c 10 -w test.pcap

  Capture the data into a file name test. packet capture

  tcpdump -r test.pcap: it open the file

- Wireshark see video of CN lab after 1hr10min.
  - icmp - **packet visualisation**
  - ip.src == 192.168.0.124
  - http - tcp stream (and not udp)

▼ NAT

Network Address Translation (NAT) - GeeksforGeeks

To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a

https://www.geeksforgeeks.org/network-address-translation-nat/

▼ Ports

PORT NUMBER	TRANSPORT PROTOCOL	SERVICE NAME
20, 21	TCP	File Transfer Protocol (FTP)
22	TCP and UDP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Simple Mail Transfer Protocol (SMTP)
53	TCP and UDP	Domain Name Server (DNS)
67, 68	UDP	Dynamic Host Configuration Protocol (DHCP)
69	UDP	Trivial File Transfer Protocol (TFTP)
80	TCP	HyperText Transfer Protocol (HTTP)

110	TCP	Post Office Protocol (POP3)
119	TCP	Network News Transport Protocol (NNTP)
123	UDP	Network Time Protocol (NTP)
135-139	TCP and UDP	NetBIOS
143	TCP and UDP	Internet Message Access Protocol (IMAP4)
161, 162	TCP and UDP	Simple Network Management Protocol (SNMP)
179	TCP	Border Gateway Protocol (BGP)
389	TCP and UDP	Lightweight Directory Access Protocol
443	TCP and UDP	HTTP with Secure Sockets Layer (SSL)

▼ VOIP

Check VoIP pdf.

When an employee calls a customer, they pick up the handset and dial them just as they normally would. The IP phone (or app) travels through your Local Area Network (LAN) switch and business router before reaching the VoIP service provider. From there, the VoIP provider establishes the call.

Bits transfered

These calls are transmitted at a fixed rate of 64 kilobits per second (Kbps) in each direction, for a total transmission rate of 128 Kbps. Since there are 8 kilobits (Kb) in a kilobyte (KB), this translates to a transmission of 16 KB each second the circuit is open, and 960 KB every minute it's open. In a 10-minute conversation, the total transmission is 9,600 KB, which is roughly equal to 10 megabytes (check out How Bits and Bytes Work

to learn about these conversions). If you look at a typical phone conversation, much of this transmitted data is wasted.

SIP

Know your protocols: VoIP protocols that work together with SIP

When we talk about VoIP, the most prominent and widely used protocol is Session Initiation Protocol (SIP). But SIP doesn't operate alone. There are many other protocols involved in IP telephony that function

http://info.teledynamics.com/blog/know-your-protocols-voip-protocols-that-work-together-with-sip

Introducing SIP Methods (VoIP Deployment)

VoIP uses only a few SIP methods today, but more methods are being released while the technology and market evolve. SIP is also a key element in transmitting other forms of non-verbal

https://what-when-how.com/voip-deployment/introducing-sip-methods-voip-deployment/

Complete website

Jumping In to the OSI Model (VoIP Deployment)

You need a basic understanding of how networks function if you want to grasp all the potential sources of latency. Fortunately, network engineers created a theoretical structure that

http://what-when-how.com/voip-deployment/jumping-in-to-the-osi-model-voip-deployment

Detailed website

Voice Over IP : Protocols and Standards

Rakesh Arora, arora@cse.wustl.edu This paper first discusses the key issues that inhibit Voice over IP (VOIP) to be popular with the users. Then I discuss the protocols and standards that exist

https://www.cse.wustl.edu/~jain/cis788-99/ftp/voip_protocols/

Frame relay

How does Frame Relay Work? - GeeksforGeeks

Frame Relay is a packet-switching network protocol that is designed to work at the data link layer of the network. It is used to connect Local Area Networks (LANs) and transmit data across

https://www.geeksforgeeks.org/how-does-frame-relay-work/