***Computer Network Lab***

**Q1. To understand various Networking Devices**

**Types of Network Devices**

Various types of network devices operate in a distinct network segment and perform distinct operations.

**1) Repeater** – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy it bit by bit and regenerate it at its star topology connectors connecting with original strength. It is a 2-port device.

**2) Hub-** A hub is a physical-layer device that acts on individual bits rather than frames. When a bit, representing a zero or a one, arrives from one interface, the hub simply recreates the bit, boosts its energy strength, and transmits the bit into all the other interfaces. Whenever a hub receives a bit from one of its interfaces, it sends a copy to all other interfaces.

**Types of Hubs**

There are generally three types of hubs that are given below.

**Active Hub :** These hubs have their power source and can clean, enhance, and relay the network's signal. It functions as both a repeater and a wiring center. The active hub may repair damaged packets as they are being sent and can also hold the direction of the remaining packets and distribute them. If a port gets a weak but readable signal, the active hub reconstructs the weak signal into a more robust signal before distributing it to other ports. If any connecting device in the network is not operating, it can increase the signal.

**Passive Hub :** The passive hubs are the wire connection points that aid in the construction of the physical network. It can detect faults and malfunctioning hardware . The passive hub accepts the packet through a port and distributes it to all ports. These hubs do not clean or enhance signals before relaying them to the network and cannot be utilized to extend the distance between nodes.

**Intelligent Hub :** It functions similarly to active hubs and offers remote management capabilities. They also supply network devices with variable data speeds. It also allows an administrator to monitor traffic flowing through the hub and manage each port.

**3) Bridge** – A bridge operates at the data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of the source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

**Types of Bridges**

**Transparent Bridges:-** These are the bridges in which the stations are completely unaware of the bridge’s existence i.e. whether or not a bridge is added or deleted from the network, reconfiguration of the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

**Source Routing Bridges -** In these bridges, routing operation is performed by the source station and the frame specifies which route to follow. The host can discover the frame by sending a special frame called the discovery frame, which spreads through the entire network using all possible paths to the destination.

**4) Switch -** A switch is a multiport bridge with a buffer and a design that can boost its efficiency(a large number of ports imply less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, which makes it very efficient as it does not forward packets that have errors and forward good packets selectively to the correct port only. In other words, the switch divides the collision domain of hosts, but the broadcast domain remains the same.

**5) Routers -** A router is a device like a switch that routes data packets based on their IP addresses. The router is mainly a Network Layer device. Routers normally connect LANs and WANs and have a dynamically updating routing table based on which they make decisions on routing the data packets. The router divides the broadcast domains of hosts connected through it.

**6. Gateway –** A gateway, as the name suggests, is a passage to connect two networks that may work upon different networking models. They work as messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switches or routers. A gateway is also called a protocol converter.

**7. Brouter –** It is also known as the bridging router is a device that combines features of both bridge and router. It can work either at the data link layer or a network layer. Working as a router, it is capable of routing packets across networks and working as the bridge, it is capable of filtering local area network traffic.

**8. NIC –** NIC or network interface card is a network adapter that is used to connect the computer to the network. It is installed in the computer to establish a LAN. It has a unique id that is written on the chip, and it has a connector to connect the cable to it. The cable acts as an interface between the computer and the router or modem. NIC card is a layer 2 device which means that it works on both the physical and data link layers of the network model.

**Q2. Understanding Basic networking Commands: ifconfig ,ip,traceroute, tracepath, ping, netstat, ss, dig, nslookup, route,host, arp, hostname, curl or wget, mtr, whois, tcpdump**

$ ifconfig

$ traceroute youtube.com

$ ping -c 5 google.com

$ tracepath youtube.com

$ ss

$ dig

$ nslookup google.com

$ route

$ host google.com OR $ host

$ hostname

$ curl google.com

$ whois youtube.com

$ sudo tcpdump

**Q3. Installation and configuration of NS2 / Introduction to Tcl Hello Programming**

sudo apt-get install gcc g++ python python-dev mercurial bzr gdb valgrind gsl-bin libgsl0-dev libgsl0ldbl flex bison tcpdump sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev libgtk2.0-0 libgtk2.0-dev uncrustify doxygen graphviz imagemagick texlive texlive-latex-extra texlive-generic-extra texlive-generic-recommended

texinfo dia texlive texlive-latex-extra texlive-extra-utils texlive-generic-recommended texi2html python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev python-pygccxml

**Q4. To create a simple simulation using NS2 with two nodes**

#Create a simulator object

set ns [new Simulator]

#Open the nam trace file

set nf [open out.nam w]

$ns namtrace-all $nf

#Define a ‘finish’ procedure

proc finish {} {

global ns nf

$ns flush-trace

#Close the trace file

close $nf

#Execute nam on the trace file

exec nam out.nam &

exit 0

}

#Create two nodes

set n0 [$ns node]

set n1 [$ns node]

#Create a duplex link between the nodes

$ns duplex-link $n0 $n1 1Mb 10ms DropTail

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 “finish”

#Run the simulation

$ns run

**Q5. Graphical Simulation using NAM(Network Animator) /**

**To create a simple simulation using NS2 with four nodes**

#Create a simulator object

set ns [new Simulator]

#Define different colors for data flows (for NAM)

$ns color 1 Blue

$ns color 2 Red

#Open the NAM trace file

set nf [open out.nam w]

$ns namtrace-all $nf

#Define a 'finish' procedure

proc finish {} {

global ns nf

$ns flush-trace

#Close the NAM trace file

close $nf

#Execute NAM on the trace file

exec nam out.nam &

exit 0

}

#Create four nodes

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

#Create links between the nodes

$ns duplex-link $n0 $n2 2Mb 10ms DropTail

$ns duplex-link $n1 $n2 2Mb 10ms DropTail

$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10

$ns queue-limit $n2 $n3 10

#Give node position (for NAM)

$ns duplex-link-op $n0 $n2 orient right-down

$ns duplex-link-op $n1 $n2 orient right-up

$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)

$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection

set tcp [new Agent/TCP]

$tcp set class\_ 2

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n3 $sink

$ns connect $tcp $sink

$tcp set fid\_ 1

#Setup a FTP over TCP connection

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type\_ FTP

#Setup a UDP connection

set udp [new Agent/UDP]

$ns attach-agent $n1 $udp

set null [new Agent/Null]

$ns attach-agent $n3 $null

$ns connect $udp $null

$udp set fid\_ 2

#Setup a CBR over UDP connection

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type\_ CBR

$cbr set packet\_size\_ 1000

$cbr set rate\_ 1mb

$cbr set random\_ false

#Schedule events for the CBR and FTP agents

$ns at 0.1 "$cbr start"

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"

$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)

$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Print CBR packet size and interval

puts "CBR packet size = [$cbr set packet\_size\_]"

puts "CBR interval = [$cbr set interval\_]"

#Run the simulation

$ns run

**Q6. Topology**

1. **Star Topology**

#Create a simulator object

set ns [new Simulator]

#Open the nam trace file

set nf [open out.nam w]

$ns namtrace-all $nf

#Define a 'finish' procedure

proc finish {} {

global ns nf

$ns flush-trace

#Close the trace file

close $nf

#Executenam on the trace file

exec nam out.nam &

exit 0

}

#Create four nodes

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

#Change the shape of center node in a star topology

$n0 shape square

#Create links between the nodes

$ns duplex-link $n0 $n1 1Mb 10ms DropTail

$ns duplex-link $n0 $n2 1Mb 10ms DropTail

$ns duplex-link $n0 $n3 1Mb 10ms DropTail

$ns duplex-link $n0 $n4 1Mb 10ms DropTail

$ns duplex-link $n0 $n5 1Mb 10ms DropTail

#Create a TCP agent and attach it to node n0

set tcp0 [new Agent/TCP]

$tcp0 set class\_ 1

$ns attach-agent $n1 $tcp0

#Create a TCP Sink agent (a traffic sink) for TCP and attach it to node n3

set sink0 [new Agent/TCPSink]

$ns attach-agent $n3 $sink0

#Connect the traffic sources with the traffic sink

$ns connect $tcp0 $sink0

# Create a CBR traffic source and attach it to tcp0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.01

$cbr0 attach-agent $tcp0

#Schedule events for the CBR agents

$ns at 0.5 "$cbr0 start"

$ns at 4.5 "$cbr0 stop"

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Run the simulation

$ns run

1. **Ring Topology**

#Create a simulator object

set ns [new Simulator]

#Open the nam trace file

set nf [open out.nam w]

$ns namtrace-all $nf

#Define a 'finish' procedure

proc finish {} {

global ns nf

$ns flush-trace

#Close the trace file

close $nf

#Executenam on the trace file

exec nam out.nam &

exit 0

}

#Create four nodes

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

#Change the shape of center node in a star topology

$n0 shape square

#Create links between the nodes

$ns duplex-link $n0 $n1 1Mb 10ms DropTail

$ns duplex-link $n1 $n2 1Mb 10ms DropTail

$ns duplex-link $n2 $n3 1Mb 10ms DropTail

$ns duplex-link $n3 $n4 1Mb 10ms DropTail

$ns duplex-link $n4 $n5 1Mb 10ms DropTail

$ns duplex-link $n5 $n0 1Mb 10ms DropTail

#Create a TCP agent and attach it to node n0

set tcp0 [new Agent/TCP]

$tcp0 set class\_ 1

$ns attach-agent $n1 $tcp0

#Create a TCP Sink agent (a traffic sink) for TCP and attach it to node n3

set sink0 [new Agent/TCPSink]

$ns attach-agent $n4 $sink0

#Connect the traffic sources with the traffic sink

$ns connect $tcp0 $sink0

# Create a CBR traffic source and attach it to tcp0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.01

$cbr0 attach-agent $tcp0

#Schedule events for the CBR agents

$ns at 0.5 "$cbr0 start"

$ns at 4.5 "$cbr0 stop"

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Run the simulation

$ns run

1. **Bus/Hub Topology**

#Create a simulator object

set ns [new Simulator]

#Open the nam trace file

set nf [open out.nam w]

$ns namtrace-all $nf

#Define a 'finish' procedure

proc finish {} {

global ns nf

$ns flush-trace

#Close the trace file

close $nf

#Executenam on the trace file

exec nam out.nam &

exit 0

}

#Create four nodes

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

set n6 [$ns node]

set n7 [$ns node]

#Change the shape of center node in a star topology

$n0 shape square

#Create links between the nodes

$ns duplex-link $n0 $n1 1Mb 10ms DropTail

$ns duplex-link $n1 $n2 1Mb 10ms DropTail

$ns duplex-link $n2 $n3 1Mb 10ms DropTail

$ns duplex-link $n3 $n4 1Mb 10ms DropTail

$ns duplex-link $n1 $n5 1Mb 10ms DropTail

$ns duplex-link $n2 $n6 1Mb 10ms DropTail

$ns duplex-link $n3 $n7 1Mb 10ms DropTail

#Create a TCP agent and attach it to node n0

set tcp0 [new Agent/TCP]

$tcp0 set class\_ 1

$ns attach-agent $n0 $tcp0

#Create a TCP Sink agent (a traffic sink) for TCP and attach it to node n3

set sink0 [new Agent/TCPSink]

$ns attach-agent $n4 $sink0

#Connect the traffic sources with the traffic sink

$ns connect $tcp0 $sink0

# Create a CBR traffic source and attach it to tcp0

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.01

$cbr0 attach-agent $tcp0

#Schedule events for the CBR agents

$ns at 0.5 "$cbr0 start"

$ns at 4.5 "$cbr0 stop"

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Run the simulation

$ns run

**Q7. To analyze the performance of a network for QoS (quality of service)**

**parameters.**

#Wireless Network

#Setting values for variables in an associative array: val

#chan: Channel Type: Wireless

#prop: Radio Propagation Model: Two way propagation

#netif: Network Interface Types: Wireless

#mac: MAC type: Cellular Communication

#ifq: Interface Queue type

#ll: Link Layer type

#ant: Antenna Type

#ifqlen: Interface queue length

#nn: Number of nodes

#rp: ad-hoc routing protocol: Destination Sequenced Distance Vector (DSDV)

#X, Y: Positions

#Stop: Stop time

set val(chan) Channel/WirelessChannel

set val(prop) Propagation/TwoRayGround ;

set val(netif) Phy/WirelessPhy ;

set val(mac) Mac/802\_11 ;

set val(ifq) Queue/DropTail/PriQueue ;

set val(ll) LL ;

set val(ant) Antenna/OmniAntenna ;

set val(ifqlen) 50 ;

set val(nn) 3 ;

set val(rp) DSDV ;

set val(x) 500 ;

set val(y) 400 ;

set val(stop) 150 ;

#Create simulator object and link the trace files and nam trace

set ns [new Simulator]

set tracefd [open simple-dsdv.tr w]

set namtrace [open simwrls.nam w]

#Linking trace files to trace buffers

$ns trace-all $tracefd

$ns use-newtrace

$ns namtrace-all-wireless $namtrace $val(x) $val(y)

#Create topography flatgrid refers to movement in XY plane

set topo [new Topography]

$topo load\_flatgrid $val(x) $val(y)

#Create General Operations Director(GOD) object

#GOD object stores total number of mobile nodes & table of shortest hops required

create-god $val(nn)

#Configuring nodes

# agentTrace: tracing at agent level turned ON or OFF

# routerTrace: tracing at router level turned ON or OFF

# macTrace: tracing at mac level turned ON or OFF

$ns node-config -adhocRouting $val(rp) \

-llType $val(ll) \

-macType $val(mac) \

-ifqType $val(ifq) \

-ifqLen $val(ifqlen) \

-antType $val(ant) \

-propType $val(prop) \

-phyType $val(netif) \

-channelType $val(chan) \

-topoInstance $topo \

-agentTrace ON \

-routerTrace ON \

-macTrace OFF \

-movementTrace ON

#Creating three nodes

#Setting node positions, z=0 as topology is flatgrid

for {set i 0} {$i < $val(nn) } { incr i } {

set node\_($i) [$ns node]

}

#Setting node mobility

$node\_(0) set X\_ 5.0

$node\_(0) set Y\_ 5.0

$node\_(0) set Z\_ 0.0

$node\_(1) set X\_ 490.0

$node\_(1) set Y\_ 285.0

$node\_(1) set Z\_ 0.0

$node\_(2) set X\_ 150.0

$node\_(2) set Y\_ 240.0

$node\_(2) set Z\_ 0.0

# setdest params: X, Y, speed.

$ns at 10.0 "$node\_(0) setdest 250.0 250.0 3.0"

$ns at 15.0 "$node\_(1) setdest 45.0 285.0 5.0"

$ns at 110.0 "$node\_(0) setdest 480.0 300.0 5.0"

#Attaching transport layer protocol agents and application layer protocol agents to the nodes

#Setting source node, sink node

set tcp [new Agent/TCP/Newreno]

set sink [new Agent/TCPSink]

$ns attach-agent $node\_(0) $tcp

$ns attach-agent $node\_(1) $sink

$ns connect $tcp $sink

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ns at 10.0 "$ftp start"

#defines Node initial position. 30 is node size Must be called after mobility

for {set i 0} {$i < $val(nn)} { incr i } {

$ns initial\_node\_pos $node\_($i) 30

}

#Reset positions at stop

for {set i 0} {$i < $val(nn) } { incr i } {

$ns at $val(stop) "$node\_($i) reset";

}

#Start and stop wireless simulation

$ns at $val(stop) "$ns nam-end-wireless $val(stop)"

$ns at $val(stop) "stop"

$ns at 150.01 "puts \"end simulation\" ; $ns halt"

proc stop {} {

#Flush trace buffers, close files and execute nam file

global ns tracefd namtrace

$ns flush-trace

close $tracefd

close $namtrace

exec nam simwrls.nam &

exit

}

$ns run

**2. Getting Throughput**

BEGIN {

recvdSize = 0 # received packet size

startTime = 400 # high random start time

stopTime = 5 #low random stop time

}

{

#Analyse the trace file

event = $1 # send or received (s/r)

time = $3 # time of transaction (time of sending)

pkt\_size = $37 # packet size

level = $19 # application agent or routing protocol data (AGT/RTR)

# Find the starting time of simulation

if (level == "AGT" && event == "s" ){

if (time < startTime){

startTime = time;

}

}

# Update total received packet size and store packets arrival time, to finally get the

# end time of the simulation

if (level == "AGT" && event == "r" ){

if (time > stopTime){

stopTime = time;

}

recvdSize += pkt\_size;

}

}

END{

# calculate the throughput

printf("Average Throughput[kbps] = %.2f\n",(recvdSize/(stopTime-startTime)))

}

**3. Getting Packet delivery ratio**

BEGIN {

sendLine = 0;

recvLine = 0;

}

$0 ~/^s.\* AGT/ {

sendLine ++ ;

}

$0 ~/^r.\* AGT/ {

recvLine ++ ;

}

END {

printf "cbr s:%d r:%d, r/s Ratio:%.4f \n", sendLine, recvLine,

(recvLine/sendLine);

}

**4. Number of dropped packets**

BEGIN{

countDropped = 0 ;

}

$0~/^d/{

countDropped++;

}

END{

printf"cbr Count of dropped packets:%d\n",countDropped;

}

**Q8. To use Wireshark to analyze the incoming of packets**

Wireshark Installation:

Wireshark Home Page:

Wireshark Capturing from Wifi window:

Wireshark ip.src\_host filter applied:

Wireshark ip.addr filter applied:

Wireshark ip.dst filter applied:

Wireshark tcp filter applied:

Wireshark udp filter applied:

Wireshark http filter applied:

**+ What is wireshark?**

Wireshark is a free and open-source network protocol analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. It captures and displays network traffic on a network interface in real-time and allows users to examine the traffic at a microscopic level. Wireshark can analyze protocols like HTTP, FTP, DNS, TCP, UDP, and many others, and can be used to detect network problems, identify network security issues, and debug network applications. It runs on multiple platforms, including Windows, macOS, and Linux.

**+ Computer aur printer kaunse topology se connected hai college lab mein ?**

**+ OSI ke baare mein / + What is the OSI model and why is it important in computer networks?**

The OSI (Open Systems Interconnection) model is a conceptual framework that defines how communication between devices in a network should be carried out. It consists of seven layers, each responsible for a specific set of functions, such as physical transmission, data routing, and application-level protocols.

* Physical Layer: Defines the physical characteristics of the transmission medium, such as the electrical and mechanical properties of cables and connectors.
* Data Link Layer: Provides error-free transfer of data between adjacent nodes on a network, using protocols such as Ethernet or Wi-Fi.
* Network Layer: Provides logical addressing and routing of data across multiple networks, ensuring that data is sent to the correct destination.
* Transport Layer: Provides end-to-end communication between applications on different hosts, ensuring that data is delivered reliably and in the correct order.
* Session Layer: Establishes, manages, and terminates sessions between applications, providing synchronization and recovery functions.
* Presentation Layer: Responsible for data representation and conversion, ensuring that data is presented in a format that is meaningful to the application layer.
* Application Layer: Provides services directly to the user or application, such as email, file transfer, or web browsing.

**IMPORTANCE**:

The OSI model is important in computer networks because it provides a standard reference model that enables different devices and systems to communicate with each other. It allows network engineers and developers to understand the functions and interactions of different network protocols and technologies, and to troubleshoot and debug network problems more effectively. The OSI model also enables interoperability between different vendors' products and facilitates the development of new network technologies and protocols.

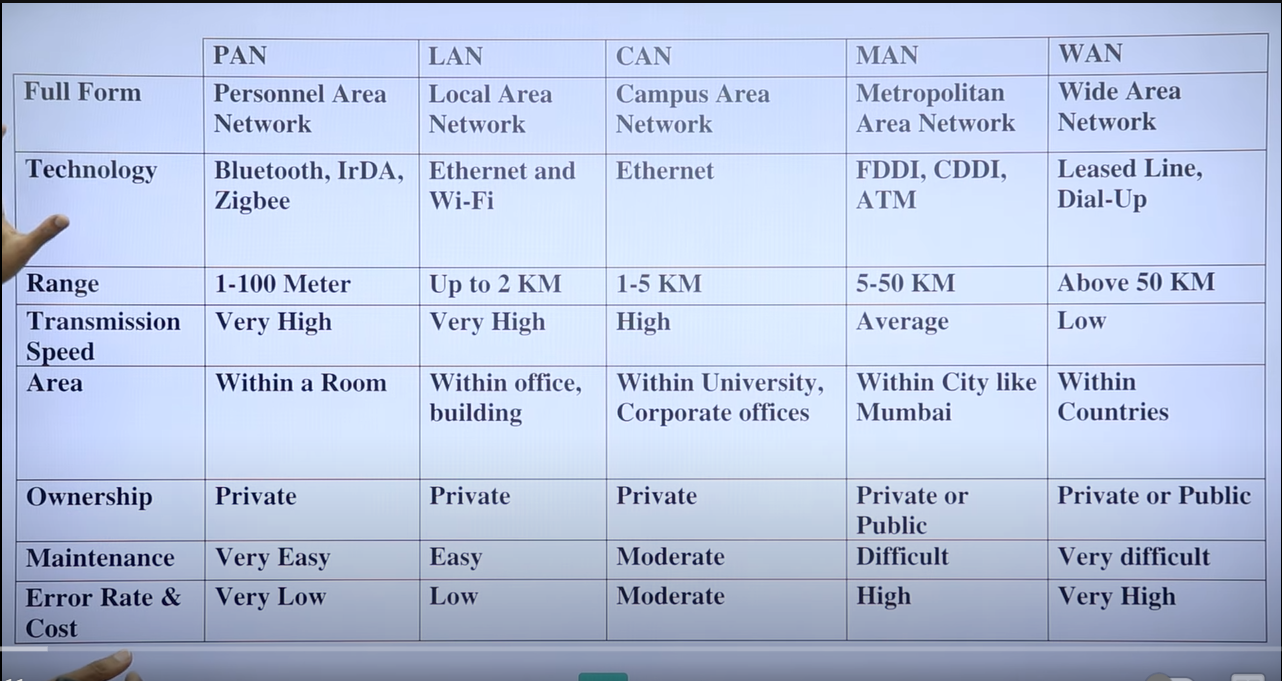
**+ What is a computer network and why is it important?**

A computer network is a collection of computers, servers, printers, and other devices that are connected together to share resources and communicate with each other. Networks can be connected using wired or wireless technologies, and can range in size from a small home network to a large enterprise-level network spanning multiple locations.

Computer networks are important because they enable efficient communication and resource sharing between devices. They allow users to access and share data, files, and applications across different devices and locations. They also enable collaboration and teamwork by allowing multiple users to work on the same project or document simultaneously. Additionally, computer networks provide a platform for internet connectivity, enabling users to access online services and information.

In a business environment, computer networks are essential for efficient operation and productivity. They facilitate communication between departments and enable employees to work remotely or from different locations. Networks also provide a platform for centralized data storage, which can be backed up and secured, ensuring data integrity and availability.

**+ What is the difference between LAN, MAN, and WAN?**

****

LAN (Local Area Network): A LAN is a network that is confined to a small geographic area, such as a home, office, or campus. It typically connects devices within a single building or location and is usually owned and managed by a single organization. LANs are characterized by high-speed data transfer rates and low latency, making them ideal for applications such as file sharing, printing, and online gaming.

MAN (Metropolitan Area Network): A MAN is a network that spans a larger geographic area than a LAN but is still limited to a specific region, such as a city or town. It typically consists of multiple LANs connected by high-speed links, such as fiber optic cables. MANs are designed to provide high-speed connectivity over a larger area than a LAN, making them suitable for applications such as video conferencing, online streaming, and large-scale file sharing.

WAN (Wide Area Network): A WAN is a network that spans a wide geographic area, such as a country, continent, or the entire world. It typically consists of multiple LANs and MANs connected over long distances using technologies such as leased lines, satellite links, or the internet. WANs are designed to provide connectivity over large distances and are suitable for applications such as online banking, e-commerce, and cloud computing.

**+ What is TCP/IP and how does it relate to the OSI model?**

TCP (Transmission Control Protocol) is responsible for ensuring reliable, ordered delivery of data between applications running on different devices. It manages the flow of data by establishing and maintaining connections between devices, breaking data into packets for transmission, and reassembling packets at the receiving end.

IP (Internet Protocol) is responsible for providing logical addressing and routing of data across networks. It determines the destination of packets and the best route for them to take across the network, using addressing schemes such as IP addresses.

While the OSI model has seven layers, the TCP/IP model has four layers:

Network Access Layer: Similar to the Physical and Data Link Layers in the OSI model, this layer is responsible for establishing and maintaining physical connections between devices, such as Ethernet or Wi-Fi.

Internet Layer: Similar to the Network Layer in the OSI model, this layer is responsible for logical addressing and routing of data across networks, using IP addressing.

Transport Layer: Similar to the Transport Layer in the OSI model, this layer is responsible for ensuring reliable, ordered delivery of data between applications running on different devices, using protocols such as TCP or UDP (User Datagram Protocol).

Application Layer: Similar to the Application Layer in the OSI model, this layer provides services directly to the user or application, such as HTTP (Hypertext Transfer Protocol) for web browsing, SMTP (Simple Mail Transfer Protocol) for email, and FTP (File Transfer Protocol) for file transfer.

**+ What is a protocol and why is it important in computer networks?**

A protocol is a set of rules and standards that dictate how devices should communicate and exchange data in a network. It is important in computer networks because it ensures that data is transmitted in a reliable, standardized, and secure manner, allowing different devices and systems to communicate with each other effectively.

Protocols are essential in computer networks as they provide a standardized set of rules and procedures for devices to communicate and exchange data in a reliable and secure manner. They enable interoperability between different devices and systems, ensure efficient data transmission, and enable effective troubleshooting and debugging of network issues.

**+ What is a switch and how does it differ from a hub?**

A switch is a network device that connects devices together in a Local Area Network (LAN) and directs data traffic between them. Unlike a hub, a switch operates at the Data Link Layer (Layer 2) of the OSI model and uses MAC (Media Access Control) addresses to identify and direct data packets to the correct destination device. This enables switches to provide higher bandwidth and faster data transfer speeds compared to hubs, which operate at the Physical Layer (Layer 1) and simply broadcast data to all connected devices.

**+ What is a router and how does it work in a network?**

A router is a network device that connects multiple networks together and directs data traffic between them. It operates at the Network Layer (Layer 3) of the OSI model and uses IP (Internet Protocol) addresses to identify and route data packets to the correct destination network. Routers use routing tables and algorithms to determine the best path for data to travel across different networks, enabling them to connect networks and provide internet connectivity to devices.

**+ What is a firewall and what is its role in network security?**

A firewall is a network security device that monitors and controls incoming and outgoing network traffic based on predefined security rules. Its role is to prevent unauthorized access to or from a network while allowing legitimate traffic to pass through. Firewalls use various methods such as packet filtering, stateful inspection, and application-level gateway filtering to analyze network traffic and prevent unauthorized access, protecting networks from malicious attacks and unauthorized access.

**+ What is NAT and how does it work in a network?**

NAT (Network Address Translation) is a technique used in computer networking that allows devices with private IP addresses to communicate with devices on the internet. NAT operates on a router or firewall and works by translating private IP addresses used on the internal network into a single public IP address that is used to communicate with devices on the internet. This helps to conserve public IP addresses and provides an additional layer of security by hiding the internal network's IP addresses from external devices. NAT can be configured to allow inbound traffic from the internet to access specific devices on the internal network, while still keeping the network secure by not exposing all devices to the internet.

**+ What is DNS and how does it work in a network?**

DNS (Domain Name System) is a hierarchical naming system that is used to translate domain names into IP addresses. In a network, DNS servers act as a directory that translates user-friendly domain names, such as www.example.com, into the IP addresses that computers use to identify each other on the internet. When a user types a domain name into their browser, the browser sends a request to a DNS server to obtain the IP address associated with that domain name. DNS servers are essential for internet communication and enable users to access websites using easy-to-remember domain names instead of complex IP addresses.

**+ What is DHCP and how does it work in a network?**

DHCP (Dynamic Host Configuration Protocol) is a network protocol used to dynamically assign IP addresses and network configuration settings to devices on a network. In a network, a DHCP server automatically assigns IP addresses to devices that request them, ensuring that devices are properly configured to communicate on the network. DHCP also allows for the assignment of other network settings such as subnet mask, default gateway, and DNS server addresses. DHCP simplifies network administration and eliminates the need for manual IP address configuration, making it easier to manage large networks.

**+ What is a VPN and how does it work in a network?**

A VPN (Virtual Private Network) is a secure connection between a user's device and a private network over the internet. In a network, a VPN encrypts the user's internet traffic and routes it through a secure connection to a VPN server. The VPN server then decrypts the traffic and forwards it to the intended destination. VPNs protect user privacy and enhance security by hiding the user's IP address and encrypting their data, preventing unauthorized access to network resources and protecting against eavesdropping and data theft.

**+ What is wireless networking and how does it differ from wired networking?**

Wireless networking is a method of connecting devices to a network without the need for physical cables or wires. It uses radio waves or infrared signals to transmit data between devices. In contrast, wired networking requires physical cables to connect devices to the network. Wireless networking provides greater flexibility and mobility, as devices can be connected to the network from anywhere within range of the wireless signal. However, wired networking is generally faster and more reliable than wireless networking, and is often preferred for applications that require high speed or low latency.

**+ What is the difference between IPv4 and IPv6?**

IPv4 and IPv6 are two versions of the Internet Protocol used for identifying devices on a network. IPv4 uses 32-bit addresses and can support up to 4.3 billion unique addresses, while IPv6 uses 128-bit addresses and can support an almost unlimited number of unique addresses. IPv6 also includes improvements such as better security and support for new technologies, while IPv4 has limitations that have led to the need for Network Address Translation (NAT) and other workarounds to address the shortage of available addresses.

**+ What is a subnet mask and how is it used in networking?**

A subnet mask is a 32-bit number used in IP addressing that helps define the boundaries of a network. It is used in conjunction with an IP address to determine which part of the address represents the network ID and which part represents the host ID. The subnet mask consists of a series of ones followed by a series of zeros, with the number of ones indicating the number of bits used for the network ID. This helps to divide large networks into smaller subnetworks, or subnets, to improve network efficiency and security.

**+ What is a gateway and how does it work in a network?**

A gateway is a network device that connects two different networks and acts as a bridge between them. It has two network interfaces, one connected to the internal network and the other connected to the external network. The gateway forwards data packets between the two networks, examining the destination address to determine the appropriate interface for transmission. Gateways can also perform functions such as network address translation and firewalling for security purposes.

**+ What is bandwidth and how is it measured?**

Bandwidth is the maximum amount of data that can be transmitted over a network in a given amount of time. It is usually measured in bits per second (bps), kilobits per second (Kbps), megabits per second (Mbps), or gigabits per second (Gbps).

Bandwidth is determined by the capacity of the network infrastructure, including the cables, switches, and routers that make up the network. It is also affected by network congestion, which can reduce the available bandwidth and slow down data transmission.

Bandwidth can be measured using specialized network testing tools such as network analyzers and network performance monitors. These tools can measure the amount of data transmitted over a network in a given amount of time and calculate the bandwidth based on this information.

In addition to measuring overall network bandwidth, it is also important to measure the bandwidth available to individual devices on the network. This can be done using tools such as bandwidth tests, which measure the amount of data that can be transmitted between a device and a server or between two devices on the network.

In summary, bandwidth is the maximum amount of data that can be transmitted over a network in a given amount of time and is usually measured in bits per second. It is determined by the capacity of the network infrastructure and can be measured using specialized network testing tools.

**+ What is latency and how is it measured?**

Latency is the amount of time it takes for data to travel from its source to its destination on a network. It is often measured in milliseconds (ms) and is a key factor in determining network performance.

There are several factors that can contribute to latency, including network congestion, distance between devices, and the processing time required by network devices.

Latency can be measured using several different tools, including ping and traceroute. Ping sends a small packet of data to a device and measures the time it takes for the device to respond. Traceroute sends packets of data to each device along the path between the source and destination and measures the time it takes for each device to process the data.

In addition to these tools, there are also specialized network testing tools that can be used to measure latency, such as network analyzers and network performance monitors. These tools provide detailed information about network latency and can help identify the source of latency issues on a network.

In summary, latency is the amount of time it takes for data to travel from its source to its destination on a network and is often measured in milliseconds. It can be measured using tools such as ping, traceroute, and specialized network testing tools.

**+ What is a packet and how is it transmitted in a network?**

A packet is a unit of data that is transmitted over a network. It consists of a header and a payload. The header contains information such as the source and destination IP addresses, the protocol used, and other metadata. The payload contains the actual data being transmitted.

When a device sends a packet over a network, the packet is passed down through the network stack to the Data Link layer, where it is encapsulated in a frame with the MAC addresses of the source and destination devices. The frame is then transmitted over the network medium, such as Ethernet, and received by the destination device.

The destination device then examines the frame to extract the packet and processes it based on the information in the header. This process continues as packets are transmitted and received between devices on the network, allowing for the exchange of data and communication between devices.

In summary, packets are units of data that are transmitted over a network and are encapsulated in frames with the MAC addresses of the source and destination devices. They are processed by devices based on information in the packet header to enable communication and exchange of data between devices on a network.

**+ What is a MAC address and how is it used in networking?**

A MAC address is a unique identifier assigned to network interfaces for communications on a network. It is used to identify network devices on a local area network (LAN) and is used in conjunction with the Address Resolution Protocol (ARP) to translate IP addresses to MAC addresses. Devices use MAC addresses to send data to specific devices on the same network.

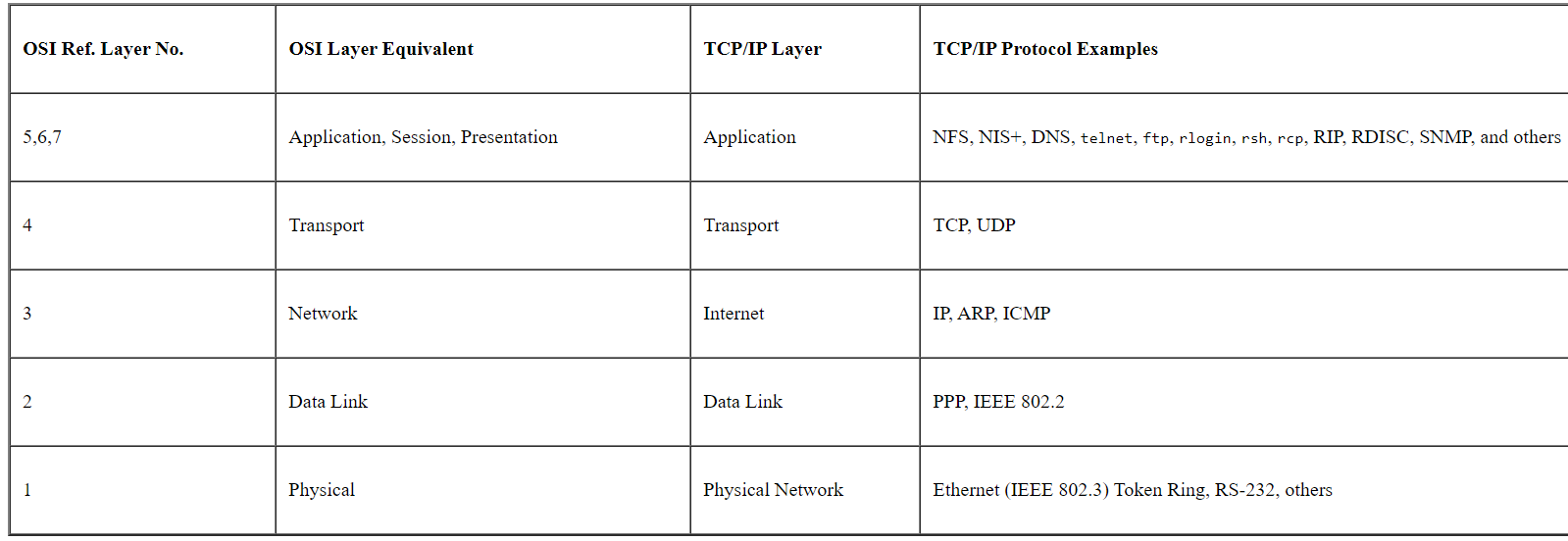
**+ In what context can we say that Wireshark is a Network analyser?**

Wireshark is a network protocol analyzer, or an application that captures packets from a network connection, such as from your computer to your home office or the internet. Packet is the name given to a discrete unit of data in a typical Ethernet network.

Wireshark is the most often-used packet sniffer in the world. Like any other packet sniffer, Wireshark does three things:

1. Packet Capture: Wireshark listens to a network connection in real time and then grabs entire streams of traffic – quite possibly tens of thousands of packets at a time.
2. Filtering: Wireshark is capable of slicing and dicing all of this random live data using filters. By applying a filter, you can obtain just the information you need to see.
3. Visualization: Wireshark, like any good packet sniffer, allows you to dive right into the very middle of a network packet. It also allows you to visualize entire conversations and network streams.

**+ Layers & unke protocols konse hai and kya karte hai Aur jo tumhe aata bata dena**

****

### Physical Network Layer

The physical network layer specifies the characteristics of the hardware to be used for the network. For example, it specifies the physical characteristics of the communications media. The physical layer of TCP/IP describes hardware standards such as IEEE 802.3, the specification for Ethernet network media, and RS-232, the specification for standard pin connectors.

### Data-Link Layer

The data-link layer identifies the network protocol type of the packet, in this case TCP/IP. It also provides error control and "framing." Examples of data-link layer protocols are Ethernet IEEE 802.2 framing and Point-to-Point Protocol (PPP) framing.

### Internet Layer

This layer, also known as the network layer, accepts and delivers packets for the network. It includes the powerful Internet protocol (IP), the Address Resolution Protocol (ARP) protocol, and the Internet Control Message Protocol (ICMP) protocol.

#### IP Protocol

The IP protocol and its associated routing protocols are possibly the most significant of the entire TCP/IP suite. IP is responsible for:

* IP addressing - The IP addressing conventions are part of the IP protocol. ([Chapter 5, Planning Your TCP/IP Network](https://docs.oracle.com/cd/E19455-01/806-0916/6ja85398p/index.html) describes IPv4 addressing in detail and [Chapter 14, Overview of IPv6](https://docs.oracle.com/cd/E19455-01/806-0916/6ja8539ba/index.html) describes IPv6 addressing in detail.)
* Host-to-host communications - IP determines the path a packet must take, based on the receiving host's IP address.
* Packet formatting - IP assembles packets into units known as IP datagrams. Datagrams are fully described in ["Internet Layer"](https://docs.oracle.com/cd/E19455-01/806-0916/6ja85398n/index.html#ipov-38).
* Fragmentation - If a packet is too large for transmission over the network media, IP on the sending host breaks the packet into smaller fragments. IP on the receiving host then reconstructs the fragments into the original packet.

Previous releases of the Solaris operating environment implemented version 4 of the Internet Protocol, which is written IPv4. However, because of the rapid growth of the Internet, it was necessary to create a new Internet Protocol with improved capabilities, such as increased address space. This new version, known as version 6, is written IPv6. The Solaris operating environment supports both versions, which are described in this book. To avoid confusion when addressing the Internet Protocol, the following convention is used:

* When the term IP is used in a description, the description applies to both IPv4 and IPv6.
* When the term IPv4 is used in a description, the description applies only to IPv4.
* When the term IPv6 is used in a description, the description applies only to IPv6.

#### ARP Protocol

The Address Resolution Protocol (ARP) conceptually exists between the data link and Internet layers. ARP assists IP in directing datagrams to the appropriate receiving host by mapping Ethernet addresses (48 bits long) to known IP addresses (32 bits long).

#### ICMP Protocol

Internet Control Message Protocol (ICMP) is the protocol responsible for detecting network error conditions and reporting on them. ICMP reports on:

* Dropped packets (when packets are arriving too fast to be processed)
* Connectivity failure (when a destination host can't be reached)
* Redirection (which tells a sending host to use another router)

The ["ping Command"](https://docs.oracle.com/cd/E19455-01/806-0916/6ja85399d/index.html) contains more information on the operating system commands that use ICMP for error detection.

### Transport Layer

The TCP/IP transport layer protocols ensure that packets arrive in sequence and without error, by swapping acknowledgments of data reception, and retransmitting lost packets. This type of communication is known as "end-to-end." Transport layer protocols at this level are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

#### TCP Protocol

TCP enables applications to communicate with each other as though connected by a physical circuit. TCP sends data in a form that appears to be transmitted in a character-by-character fashion, rather than as discreet packets. This transmission consists of a starting point, which opens the connection, the entire transmission in byte order, and an ending point, which closes the connection.

TCP attaches a header onto the transmitted data. This header contains a large number of parameters that help processes on the sending machine connect to peer processes on the receiving machine.

TCP confirms that a packet has reached its destination by establishing an end-to-end connection between sending and receiving hosts. TCP is therefore considered a "reliable, connection-oriented" protocol.

#### UDP Protocol

UDP, the other transport layer protocol, provides datagram delivery service. It does not provide any means of verifying that connection was ever achieved between receiving and sending hosts. Because UDP eliminates the processes of establishing and verifying connections, applications that send small amounts of data use it rather than TCP.

### Application Layer

The application layer defines standard Internet services and network applications that anyone can use. These services work with the transport layer to send and receive data. There are many applications layer protocols, some of which you probably already use. Some of the protocols include:

* Standard TCP/IP services such as the ftp, tftp, and telnet commands
* UNIX "r" commands, such as rlogin and rsh
* Name services, such as NIS+ and Domain Name System (DNS)
* File services, such as the NFS service
* Simple Network Management Protocol (SNMP), which enables network management
* RIP and RDISC routing protocols

#### Standard TCP/IP Services

* FTP and Anonymous FTP - The File Transfer Protocol (FTP) transfers files to and from a remote network. The protocol includes the ftp command (local machine) and the in.ftpd daemon (remote machine). FTP enables a user to specify the name of the remote host and file transfer command options on the local host's command line. The in.ftpd daemon on the remote host then handles the requests from the local host. Unlike rcp, ftp works even when the remote computer does not run a UNIX-based operating system. A user must log in to the remote computer to make an ftp connection unless it has been set up to allow anonymous FTP.  
  You can now obtain a wealth of materials from anonymous FTP servers connected to the Internet. These servers are set up by universities and other institutions to make certain software, research papers, and other information available to the public domain. When you log in to this type of server, you use the login name anonymous, hence the term "anonymous FTP servers."  
  Using anonymous FTP and setting up anonymous FTP servers is outside the scope of this manual. However, many trade books, such as The Whole Internet User's Guide & Catalog, discuss anonymous FTP in detail. Instructions for using FTP to reach standard machines are in [System Administration Guide, Volume 1](https://docs.oracle.com/docs/cd/E19455-01/805-7228/index.html). The [ftp(1)](https://docs.oracle.com/docs/cd/E19455-01/806-0624/6j9vek561/index.html) man page describes all ftp command options, including those invoked through the command interpreter. The [ftpd(1M)](https://docs.oracle.com/docs/cd/E19455-01/806-0625/6j9vfiln3/index.html) man page describes the services provided by the daemon in.ftpd.
* Telnet - The Telnet protocol enables terminals and terminal-oriented processes to communicate on a network running TCP/IP. It is implemented as the program telnet (on local machines) and the daemon in.telnet (on remote machines). Telnet provides a user interface through which two hosts can communicate on a character-by-character or line-by-line basis. The application includes a set of commands that are fully documented in the [telnet(1)](https://docs.oracle.com/docs/cd/E19455-01/806-0624/6j9vek5i5/index.html) man page.
* TFTP - The trivial file transfer protocol (tftp) provides functions similar to ftp, but it does not establish ftp's interactive connection. As a result, users cannot list the contents of a directory or change directories. This means that a user must know the full name of the file to be copied. The [tftp(1)](https://docs.oracle.com/docs/cd/E19455-01/806-0624/6j9vek5i9/index.html) man page describes the tftp command set.

#### UNIX "r" Commands

The UNIX "r" commands enable users to issue commands on their local machines that are actually carried out on the remote host that they specify. These commands include

* rcp
* rlogin
* rsh

Instructions for using these commands are in [rcp(1)](https://docs.oracle.com/docs/cd/E19455-01/806-0624/6j9vek5ef/index.html), [rlogin(1)](https://docs.oracle.com/docs/cd/E19455-01/806-0624/6j9vek5f2/index.html), and [rsh(1)](https://docs.oracle.com/docs/cd/E19455-01/806-0624/6j9vek5fa/index.html) man pages.

#### Name Services

Two name services are available from the Solaris implementation of TCP/IP: NIS+ and DNS.

* NIS+ - NIS+ provides centralized control over network administration services, such as mapping host names to IP and Ethernet addresses, verifying passwords, and so on. See [Solaris Naming Administration Guide](https://docs.oracle.com/docs/cd/E19455-01/806-1387/index.html) for complete details.
* Domain Name System - The Domain Name System (DNS) provides host names to the IP address service. It also serves as a database for mail administration. For a complete description of this service, see [Solaris Naming Administration Guide](https://docs.oracle.com/docs/cd/E19455-01/806-1387/index.html). See also the [in.named(1M)](https://docs.oracle.com/docs/cd/E19455-01/806-0625/6j9vfilo5/index.html) man page.

#### File Services

The NFS application layer protocol provides file services for the Solaris operating environment. You can find complete information about the NFS service in [Chapter 29, Solaris NFS Environment](https://docs.oracle.com/cd/E19455-01/806-0916/6ja8539fd/index.html).

#### Network Administration

The Simple Network Management Protocol (SNMP) enables you to view the layout of your network, view status of key machines, and obtain complex network statistics from graphical user interface based software. Many companies offer network management packages that implement SNMP; SunNet ManagerTM software is an example.

#### Routing Protocols

The Routing Information Protocol (RIP) and the Router Discovery Protocol (RDISC) are two routing protocols for TCP/IP networks. They are described in ["Routing Protocols"](https://docs.oracle.com/cd/E19455-01/806-0916/6ja85399s/index.html).

**+ Difference between http and https**

HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) are both protocols used for transmitting data over the internet, but HTTPS provides an additional layer of security.

HTTP is an application protocol that defines how data is transmitted between a web server and a web client (usually a web browser). It is used to send and receive data in the form of web pages, images, videos, and other types of content. HTTP uses port 80 by default.

HTTPS is a secure version of HTTP that adds an extra layer of security through the use of SSL (Secure Sockets Layer) or TLS (Transport Layer Security) encryption. HTTPS uses port 443 by default. When a user accesses a website over HTTPS, their browser and the server first establish a secure connection through a process called a handshake. The handshake verifies the identity of the server and encrypts all data exchanged between the server and the browser. This ensures that any data transmitted over the internet is protected from eavesdropping, tampering, and other forms of unauthorized access.

In summary, the main difference between HTTP and HTTPS is that HTTPS adds an extra layer of security through encryption, making it more secure for transmitting sensitive data such as passwords, credit card information, and other personal information over the internet.

**+ Difference between ipconfig and ifconfig**

`ipconfig` and `ifconfig` are both network configuration utilities, but they are used on different operating systems.

`ipconfig` is a command-line tool used in Windows operating systems to display the current configuration of the network adapters on a computer. It shows the IP address, subnet mask, and default gateway for each adapter. Additionally, it can be used to release and renew DHCP leases, flush DNS resolver cache, and perform other network-related tasks.

On the other hand, `ifconfig` is a similar command-line utility used in Unix-like operating systems (such as Linux, macOS, and BSD). It provides detailed information about the network interfaces on a system, including the IP address, netmask, and broadcast address. It can also be used to configure and modify network interfaces, set routes, and enable or disable network interfaces.

In summary, while both commands serve similar purposes, they are specific to different operating systems and have slightly different features and capabilities.

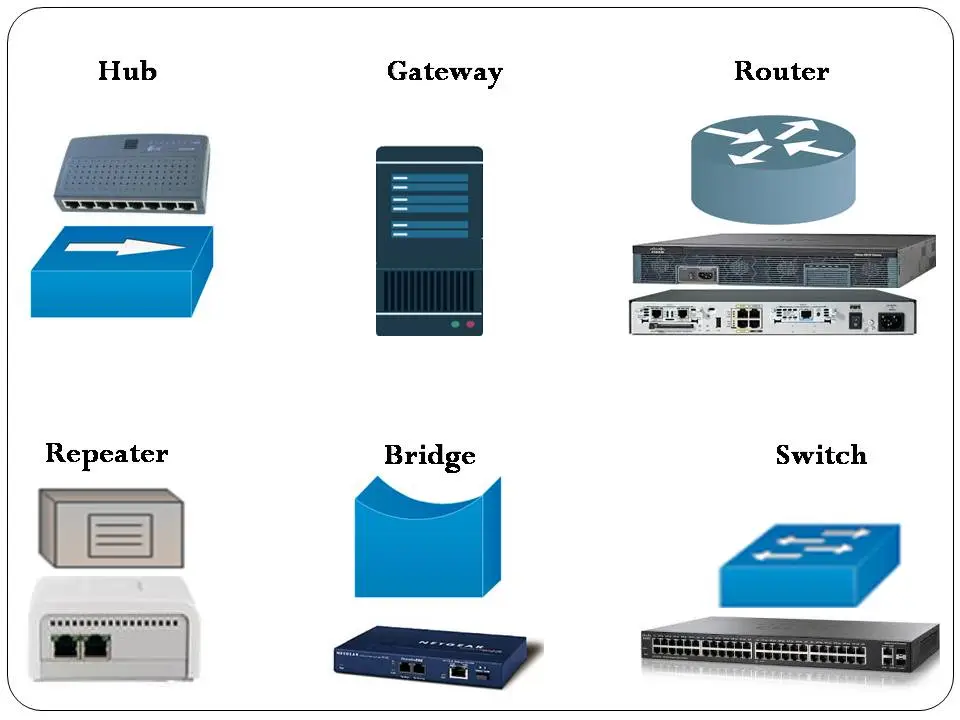
**+ What is UDP?**

User datagram protocol (UDP) operates on top of the Internet Protocol (IP) to transmit datagrams over a network. UDP does not require the source and destination to establish a three-way handshake before transmission takes place. Additionally, there is no need for an end-to-end connection.

Since UDP avoids the overhead associated with connections, error checks and the retransmission of missing data, it’s suitable for real-time or high performance applications that don’t require data verification or correction. If verification is needed, it can be performed at the application layer.

UDP is commonly used for Remote Procedure Call (RPC) applications, although RPC can also run on top of TCP. RPC applications need to be aware they are running on UDP, and must then implement their own reliability mechanisms.

**+ tumhare lab me router/bridge/switch,etc. Kaha par hai?**

****