Registration Number: 19BCE2119 **Name: Gaurav Kumar Singh**

Subject: Network and Communications (CSE1004)(L15+L16)

Digital Assignment 1

INDEX

sender→ layers(header)(insertion)

receiver→ layer(deletion)

BASIC LINUX AND NETWORK COMMANDS

1) Command: ifconfig

   Syntax: ifconfig [network_interface_name]

   ifconfig [network_interface_name] [IP/Netmask to be set]

   Execution: ifconfig, ifconfig eth1

   Output:

```
gaurav1020@DESKTOP-RORPIEK: ~
gaurav1020@DESKTOP-RORPIEK:~$ ifconfig
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 100.78.29.96  netmask 255.255.255.192  broadcast 100.78.29.127
        inet6 fe80::f47b:d7d2:3046:d688  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether 00:1e:10:1f:00:00  (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 1500
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0xfe<compat,link,site,host>
        loop  (Local Loopback)
        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

gaurav1020@DESKTOP-RORPIEK:~$ ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 100.78.29.96  netmask 255.255.255.192  broadcast 100.78.29.127
        inet6 fe80::f47b:d7d2:3046:d688  prefixlen 64  scopeid 0xfd<compat,link,site,host>
        ether 00:1e:10:1f:00:00  (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
```

2) Command: ip

   Syntax: ip [OPTIONS] OBJECT { COMMAND | help }

   ip [OPTIONS] [COMMAND] [Network_interface_name]

   Execution: ip a show eth1, ip l show eth1

   Output:

```
gaurav1020@DESKTOP-RORPIEK: ~
gaurav1020@DESKTOP-RORPIEK:~$ ip a show eth1
15: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1
    link/ether 00:1e:10:1f:00:00
    inet 100.78.29.96/26 brd 100.78.29.127 scope global dynamic
       valid_lft forever preferred_lft forever
    inet6 fe80::f47b:d7d2:3046:d688/64 scope link dynamic
       valid_lft forever preferred_lft forever
gaurav1020@DESKTOP-RORPIEK:~$ ip l show eth1
15: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 group default qlen 1
    link/ether 00:1e:10:1f:00:00
gaurav1020@DESKTOP-RORPIEK:~$
```

3) Command: traceroute

   Syntax: traceroute [IP/URL]

Execution: traceroute localhost

Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
gaurav1020@DESKTOP-R0RPIEK:~$ traceroute localhost
traceroute to localhost (127.0.0.1), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

4) Command: man

Syntax: man [EXECUTABLE_SHELL_COMMANDS/ SYSTEM_CALLS/ LIBRARY_CALLS etc.]

Execution: man tracepath {executable shell command}

Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
TRACEPATH(8)                         iputils                         TRACEPATH(8)

NAME
       tracepath - traces path to a network host discovering MTU along this path

SYNOPSIS
       tracepath [-4] [-6] [-n] [-b] [-l pktlen] [-m max_hops] [-p port] [-V] {destination}

DESCRIPTION
       It traces path to destination discovering MTU along this path. It uses UDP port port or some
       random port. It is similar to traceroute, only does not require superuser privileges and has
       no fancy options.

       tracepath -6 is good replacement for traceroute6 and classic example of application of Linux
       error queues. The situation with IPv4 is worse, because commercial IP routers do not return
       enough information in ICMP error messages. Probably, it will change, when they will be
       updated. For now it uses Van Jacobson's trick, sweeping a range of UDP ports to maintain trace
       history.

OPTIONS
       -4
           Use IPv4 only..

       -6
           Use IPv6 only..

       -n
           Print primarily IP addresses numerically.

       -b
           Print both of host names and IP addresses.

       -l
           Sets the initial packet length to pktlen instead of 65535 for IPv4 or 128000 for IPv6.

       -m
           Set maximum hops (or maximum TTLs) to max_hops instead of 30.

       -p
           Sets the initial destination port to use.

       -V
           Print version and exit.

OUTPUT
           root@mops:~ # tracepath -6 3ffe:2400:0:109::2
            1?: [LOCALHOST]                              pmtu 1500
            1:  dust.inr.ac.ru                   0.411ms
 Manual page tracepath(8) line 1 (press h for help or q to quit)
```

5) Command: ping

Back to Index

Syntax: ping [OPTION] [TARGET IP/ URL]

Execution: ping -c 10 google.com

Output:

```
gaurav1020@DESKTOP-RORPIEK: ~
gaurav1020@DESKTOP-RORPIEK:~$ ping -c 10 google.com
PING google.com (172.217.167.206) 56(84) bytes of data.
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=1 ttl=116 time=562 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=2 ttl=116 time=56.8 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=3 ttl=116 time=55.6 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=4 ttl=116 time=55.0 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=5 ttl=116 time=54.2 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=6 ttl=116 time=53.4 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=7 ttl=116 time=50.5 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=8 ttl=116 time=51.8 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=9 ttl=116 time=48.2 ms
64 bytes from del03s18-in-f14.1e100.net (172.217.167.206): icmp_seq=10 ttl=116 time=47.1 ms

--- google.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9008ms
rtt min/avg/max/mdev = 47.137/103.424/561.583/152.748 ms
gaurav1020@DESKTOP-RORPIEK:~$
```

6) Command: netstat

Syntax: netstat [OPTION]

Execution: netstat -r, netstat -i

Output:

```
gaurav1020@DESKTOP-RORPIEK: ~
gaurav1020@DESKTOP-RORPIEK:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
127.0.0.0       0.0.0.0         255.0.0.0       U         0 0          0 lo
127.0.0.1       0.0.0.0         255.255.255.255 U         0 0          0 lo
127.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 lo
224.0.0.0       0.0.0.0         240.0.0.0       U         0 0          0 lo
255.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 lo
0.0.0.0         100.78.29.65    255.255.255.255 U         0 0          0 eth1
100.78.29.64    0.0.0.0         255.255.255.192 U         0 0          0 eth1
100.78.29.96    0.0.0.0         255.255.255.255 U         0 0          0 eth1
100.78.29.127   0.0.0.0         255.255.255.255 U         0 0          0 eth1
224.0.0.0       0.0.0.0         240.0.0.0       U         0 0          0 eth1
255.255.255.255 0.0.0.0         255.255.255.255 U         0 0          0 eth1
gaurav1020@DESKTOP-RORPIEK:~$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth1      1500        0      0      0 0             0      0      0      0 BMRU
lo        1500        0      0      0 0             0      0      0      0 LRU
gaurav1020@DESKTOP-RORPIEK:~$
```

7) Command: ss

Syntax: ss [OPTION]

Execution: ss -a

Output:

```
gaurav1020@DESKTOP-RORPIEK: ~
gaurav1020@DESKTOP-RORPIEK:~$ ss -a
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Cannot open netlink socket: Protocol not supported
Netid     State      Recv-Q      Send-Q           Local Address:Port           Peer Address:Port     Process
nl        UNCONN     0           0                         rtnl:4                           *
nl        UNCONN     0           0                         rtnl:5                           *
gaurav1020@DESKTOP-RORPIEK:~$
```

8) Command: dig

Syntax: dig [TARGET IP/ URL]

Execution: dig localhost

Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
gaurav1020@DESKTOP-R0RPIEK:~$ dig localhost

; <<>> DiG 9.16.1-Ubuntu <<>> localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 52390
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;localhost.                     IN      A

;; AUTHORITY SECTION:
.                       3600    IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2021021100 1
800 900 604800 86400

;; Query time: 98 msec
;; SERVER: 202.56.215.41#53(202.56.215.41)
;; WHEN: Thu Feb 11 13:20:55 IST 2021
;; MSG SIZE  rcvd: 113

gaurav1020@DESKTOP-R0RPIEK:~$
```

9) Command: nslookup

Syntax: nslookup [TARGET IP/ URL]

Execution: nslookup google.com

Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
gaurav1020@DESKTOP-R0RPIEK:~$ nslookup google.com
Server:         202.56.215.41
Address:        202.56.215.41#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.167.206
Name:   google.com
Address: 2404:6800:4002:80d::200e

gaurav1020@DESKTOP-R0RPIEK:~$
```

10) Command: route

Syntax: route [OPTION] [COMMAND] [IP ADDRESS]

Execution: route -n

Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
gaurav1020@DESKTOP-R0RPIEK:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
127.0.0.0       0.0.0.0         255.0.0.0       U     256    0        0 lo
127.0.0.1       0.0.0.0         255.255.255.255 U     256    0        0 lo
127.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 lo
224.0.0.0       0.0.0.0         240.0.0.0       U     256    0        0 lo
255.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 lo
0.0.0.0         100.78.29.65    255.255.255.255 U     256    0        0 eth1
100.78.29.64    0.0.0.0         255.255.255.192 U     256    0        0 eth1
100.78.29.96    0.0.0.0         255.255.255.255 U     256    0        0 eth1
100.78.29.127   0.0.0.0         255.255.255.255 U     256    0        0 eth1
224.0.0.0       0.0.0.0         240.0.0.0       U     256    0        0 eth1
255.255.255.255 0.0.0.0         255.255.255.255 U     256    0        0 eth1
gaurav1020@DESKTOP-R0RPIEK:~$
```

11) Command: host

    Syntax: host [Target IP/ URL]

    Execution: host google.com

    Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
gaurav1020@DESKTOP-R0RPIEK:~$ host google.com
google.com has address 172.217.167.206
google.com has IPv6 address 2404:6800:4002:80d::200e
google.com mail is handled by 10 aspmx.l.google.com.
google.com mail is handled by 50 alt4.aspmx.l.google.com.
google.com mail is handled by 20 alt1.aspmx.l.google.com.
google.com mail is handled by 30 alt2.aspmx.l.google.com.
google.com mail is handled by 40 alt3.aspmx.l.google.com.
gaurav1020@DESKTOP-R0RPIEK:~$
```

12) Command: iwconfig

    Syntax: iwconfig [CONFIGURATION_OPTION] [network_interface_name]

    Execution: iwconfig, iwconfig eth1

    Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
gaurav1020@DESKTOP-R0RPIEK:~$ iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wifi0     no wireless extensions.

wifi1     no wireless extensions.

wifi2     no wireless extensions.

eth1      no wireless extensions.

gaurav1020@DESKTOP-R0RPIEK:~$ iwconfig eth1
eth1      no wireless extensions.

gaurav1020@DESKTOP-R0RPIEK:~$
```

13) Command: hostname

    Syntax: hostname

    Execution: hostname

Output:

14) Command: mtr

Syntax: mtr [OPTION] [TARGET_IP/ URL]

Execution: mtr -version

Output:

15) Command: whois

Syntax: whois [TARGET_IP/ URL]

Execution: whois geeksforgeeks.org

Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
gaurav1020@DESKTOP-R0RPIEK:~$ whois geeksforgeeks.org
Domain Name: GEEKSFORGEEKS.ORG
Registry Domain ID: D155653061-LROR
Registrar WHOIS Server: whois.publicdomainregistry.com
Registrar URL: http://www.publicdomainregistry.com
Updated Date: 2018-01-29T08:59:40Z
Creation Date: 2009-03-19T06:08:55Z
Registry Expiry Date: 2023-03-19T06:08:55Z
Registrar Registration Expiration Date:
Registrar: PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID: 303
Registrar Abuse Contact Email: abuse-contact@publicdomainregistry.com
Registrar Abuse Contact Phone: +1.2013775952
Reseller:
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registrant Organization: Privacy Protect, LLC (PrivacyProtect.org)
Registrant State/Province: MA
Registrant Country: US
Name Server: NS-1520.AWSDNS-62.ORG
Name Server: NS-1569.AWSDNS-04.CO.UK
Name Server: NS-245.AWSDNS-30.COM
Name Server: NS-869.AWSDNS-44.NET
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form https://www.icann.org/wicf/)
>>> Last update of WHOIS database: 2021-02-11T09:05:14Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

Access to Public Interest Registry WHOIS information is provided to assist persons in determining the co
ntents of a domain name registration record in the Public Interest Registry registry database. The data
in this record is provided by Public Interest Registry for informational purposes only, and Public Inter
est Registry does not guarantee its accuracy. This service is intended only for query-based access. You
agree that you will use this data only for lawful purposes and that, under no circumstances will you use
 this data to (a) allow, enable, or otherwise support the transmission by e-mail, telephone, or facsimil
e of mass unsolicited, commercial advertising or solicitations to entities other than the data recipient
's own existing customers; or (b) enable high volume, automated, electronic processes that send queries
or data to the systems of Registry Operator, a Registrar, or Afilias except as reasonably necessary to r
egister domain names or modify existing registrations. All rights reserved. Public Interest Registry res
erves the right to modify these terms at any time. By submitting this query, you agree to abide by this
policy.

The Registrar of Record identified in this output may have an RDDS service that can be queried for addit
ional information on how to contact the Registrant, Admin, or Tech contact of the queried domain name.
gaurav1020@DESKTOP-R0RPIEK:~$
```

16) Command: ifplugstatus

Syntax: ifplugstatus [OPTION] [network_interface_name]

Execution: ifplugstatus

Output:

```
gaurav1020@DESKTOP-R0RPIEK: ~
gaurav1020@DESKTOP-R0RPIEK:~$ ifplugstatus
eth0: link beat detected
lo: link beat detected
wifi0: link beat detected
wifi1: link beat detected
wifi2: link beat detected
eth1: link beat detected
gaurav1020@DESKTOP-R0RPIEK:~$
```

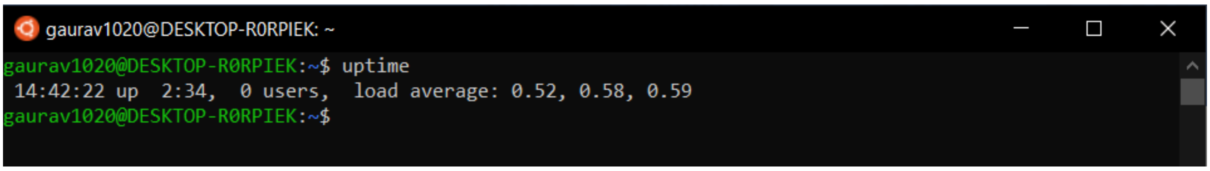
17) Command: uptime

Syntax: uptime [OPTION]

Back to Index

Execution: uptime

Output:

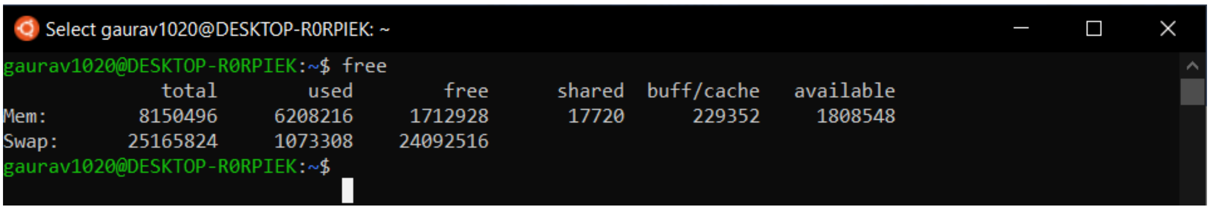


18) Command: free

Syntax: free [OPTION]

Execution: free

Output:



19) Command: ls

cat

Syntax: ls

cat [file name]

Execution: ls

cat eg1.txt

Output:

20) Command: cd

Syntax: cd [Directory name]

Execution: cd Egdir1

Output:

Back to Index

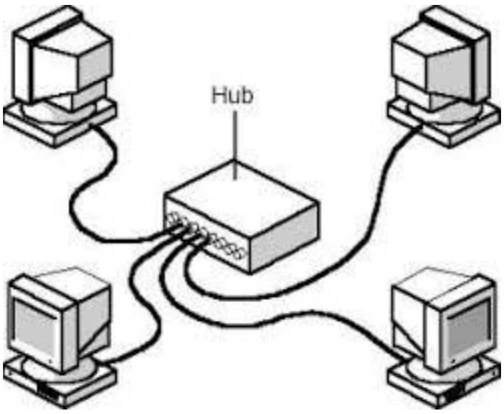
NETWORKING COMPONENTS

Some of the most widely used networking components are:-

1. Hubs

When referring to a network, a hub is the most basic networking device that connects multiple computers or other network devices together. Unlike a network switch or router, a network hub has no routing tables or intelligence on where to send information and broadcasts all network data across each connection. Most hubs can detect basic network errors, such as collisions, but having all information broadcast to multiple ports is a security risk and cause bottlenecks. In the past, network hubs were popular because they were cheaper than a switch or router. Today, switches do not cost much more than a hub and are a much better solution for any network. A hub doesn't require an IP address since it doesn't handle network traffic and can't differentiate between networks it is linking to.

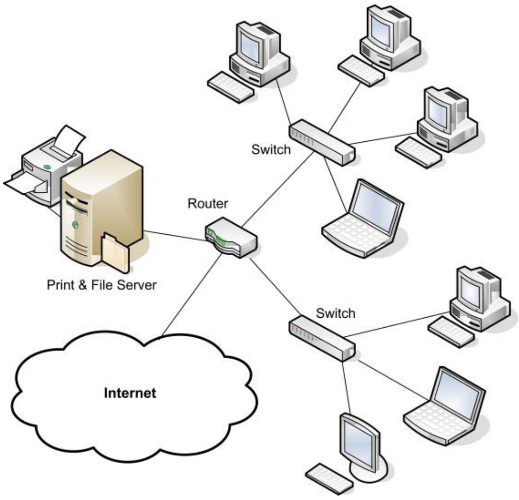
An example is a USB hub, which allows multiple USB devices to connect to one computer, even though that computer may only have a few USB connections.



2. Switches

A network switch connects devices (such as computers, printers, wireless access points) in a network to each other, and allows them to 'talk' by exchanging data packets. Switches can be hardware devices that manage physical networks, as well as software-based virtual devices.

A network switch operates on the network layer 2 of the OSI model. In a local area network (LAN) using Ethernet, a network switch determines where to send each incoming message frame by looking at the physical device address (or MAC address). Switches maintain tables that match each MAC address, to the port which the MAC address is received.



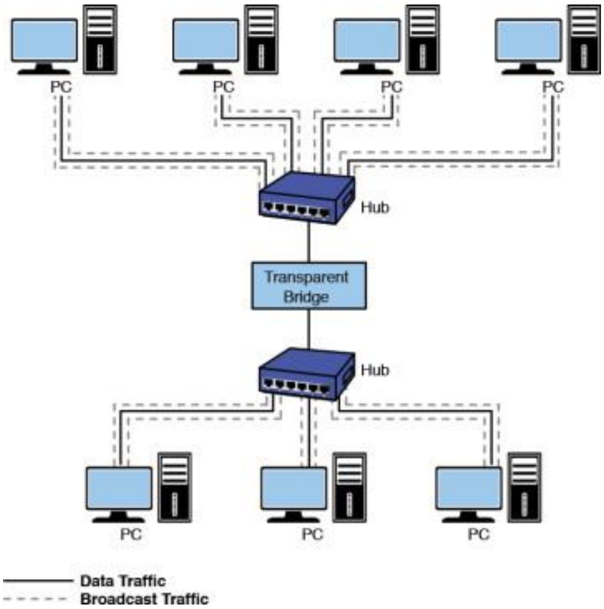
3. Bridges

A bridge is a type of computer network device that provides interconnection with other bridge networks that use the same protocol.

Bridge devices work at the data link layer of the OSI model, connecting two different networks together and providing communication between them. Bridges are similar to repeaters and hubs in that they broadcast data to every node. However, bridges maintain the MAC address table as soon as they discover new segments, so subsequent transmissions are sent to only to the desired recipient. Bridges are also known as Layer 2 switches.

A bridge functions by blocking or forwarding data, based on the destination MAC address written into each frame of data. If the bridge believes the destination address is on a network other than that from which the data was received, it can forward the data to the other networks to which it is connected. If the address is not on the other side of the bridge, the data is blocked from passing. Bridges "learn" the

MAC addresses of devices on connected networks by “listening” to network traffic and recording the

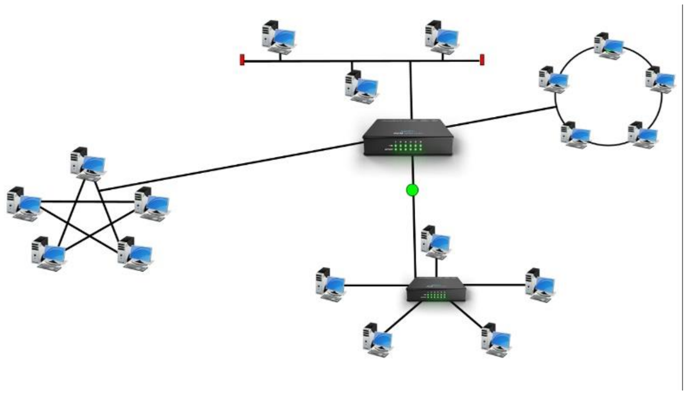


network from which the traffic originates.

4. Routers

A router is hardware device designed to receive, analyse and move incoming packets to another network. It may also be used to convert the packets to another network interface, drop them, and perform other actions relating to a network.

A router has a lot more capabilities than other network devices, such as a hub or a switch that are only able to perform basic network functions. For example, a hub can transfer data between computers or network devices but doesn't analyse or do anything with the transferred data. By contrast, routers can analyse the data sent over a network, change how it is packaged, and send it to another network or over a different network. For example, routers are commonly used in home networks to share a single Internet connection between multiple computers. The basic requirement for a router is that it must have at least two network interfaces. If they are LAN interfaces, the router can manage and route the information between two LAN segments. More commonly, a router is used to provide connectivity across wide area network (WAN) links. Unlike bridges and switches, which use the hardware-configured MAC address to determine the destination of the data, routers use the software-configured network address to make decisions. This approach makes routers more functional than bridges or switches, and it also makes them more complex because they have to work harder to determine the information.



Routers rely on two types of network protocols to make the routing magic happen:

a) Routable Protocols:- Large internetworks need protocols that allow systems to be identified by the address of the network to which they are attached and by an address that uniquely identifies them on that network. Network protocols that provide both of these features are said to be routable.
b) Routing Protocols :- are the means by which routers communicate with each other. This communication is necessary so that routers can learn the network topology and changes that occur in it

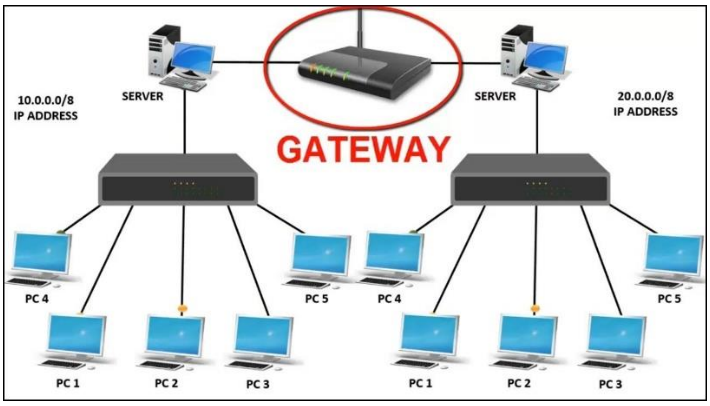
5. Gateways

Gateway is a network connecting device that can be used to connect two devices in two different networks implementing different networking protocols and overall network architecture. In other words, a gateway is a node on a network that serves as an entrance to another network. A Gateway is the most intelligent device among the network connecting devices. Intelligent in terms of its working, error control, data packet routing, transmission speed, etc. It is a combination of both hardware as well as software components.

One of the main features of using a gateway is that we can have routing controls for different networks through gateways. This way, the traffic flow in the transmission channels for different networks can be easily controlled by gateways.

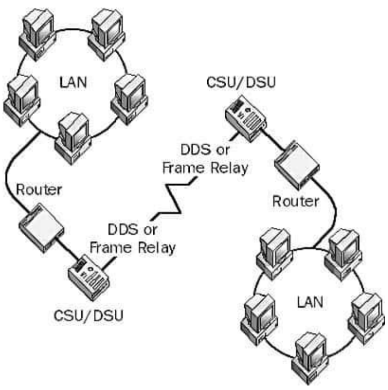
A gateway operates on all the layers of the OSI model, so it can be used as a one-stop solution for all kinds of network device connectivities. But the major disadvantage of using a gateway is its implementation cost. So, it will not be so effective to be used for small networks, or for a single network. Also, the implementation of gateways is very complex. A Gateway is also called as 'Protocol Converter' because it can convert the data packets as per the destination network protocol requirement. It can also translate the data format as per the destination needs or architecture. A gateway is used either at the starting or endpoint of the network. It is an intelligent device that can be used to connect a local node with an external node having a completely different structure(protocols/architecture/languages/data formatting structures). In other words, a gateway acts as a 'gate' between two networks and enables traffic to flow(in and out) of the network.

Gateways are often associated with both Routers and Switches. A Router routes the data packets(arriving at the gateway) to the correct node in the destination network. While a switch specifies the actual path of the data in and out of the gateway.



6. CSU/DSU

A Channel Service Unit/Data Service Unit (CSU/DSU) acts as a translator between the LAN data format and the WAN data format. Such a conversion is necessary because the technologies used on WAN links are different from those used on LANs. Some consider a CSU/DSU as a type of digital modem; but unlike a normal modem, which changes the signal from digital to analog, a CSU/DSU changes the signal from one digital format to another.



A CSU/DSU has physical connections for the LAN equipment, normally via a serial interface, and another connection for a WAN. Traditionally, the CSU/DSU has been in a separate box from other networking equipment; however, the increasing use of WAN links means that some router manufacturers are now including the CSU/DSU functionality in routers or are providing the expansion capability to do so.

7. NICs

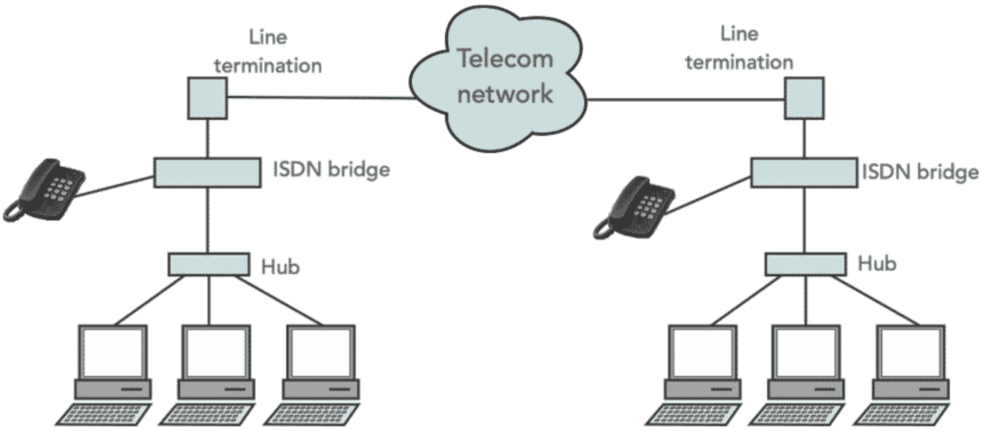
Network interface card NIC is a hardware component, where network controllers are integrated on to a circuit board that uses standard OSI model of 7 layers to communicate and it acts like a trans-receiver, where it can transmit and receives at the same time while communicating with other devices.

Network Interface Card (NIC) is a hardware unit, which is inbuilt inside a computer provided with a slot, it connects the computer to a computer network for communication with other devices via buses. There are many synonyms for network interface card like, network adapter, local area network (LAN) card or physical network interface card, ethernet controller or ethernet adapter, network controller, and connection card. Network interface card supports almost all standard buses for data transfer between the computers or devices. The connectors or buses act as an intermediator for communication converts the communication between various devices from serial communication to parallel communication or parallel communication to serial communication. It also formats data based on the architecture of the network.

8. ISDN Adapters

ISDN or Integrated Services Digital Network is an international standard for end to end digital transmission of voice, data and signalling.

ISDN can operate over copper based systems and allows the transmission of digital data over the telecommunications networks, typically ordinary copper based systems and providing higher data speeds and better quality than analogue transmission. The ISDN specifications provide a set of protocols that enable the set up, maintenance and completion of calls. ISDN is a circuit-switched telephone network that carries packets data over copper lines and enabled existing copper wire based landline technology to be used to carry digital services.



There are two types of channel that are found within ISDN:

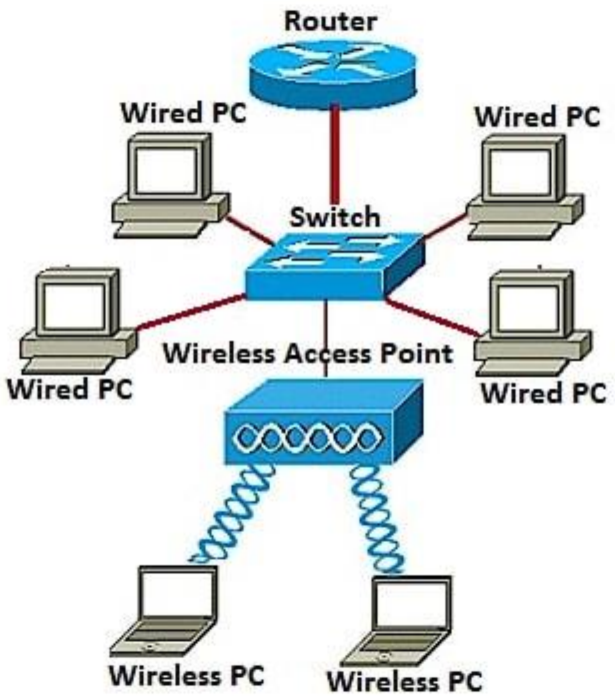
- B or Bearer channels: The bearer channels are used to carry the payload data which may be voice and / or data
- D or Delta channels: The D channels are intended for signalling and control, although it may also be used for data under some circumstances.

9. WAPs

A wireless access point (WAP) is a hardware device or configured node on a local area network (LAN) that allows wireless capable devices and wired networks to connect through a wireless standard, including Wi-Fi or Bluetooth. WAPs feature radio transmitters and antennae, which facilitate connectivity between devices and the Internet or a network.

Wireless access points (WAP) may be used to provide network connectivity in office environments, allowing employees to work anywhere in the office and remain connected to a network. In addition, WAPs provide wireless Internet in public places, like coffee shops, airports and train stations.

Wireless access points are most commonly thought of in the context of the 802 series of wireless standards, commonly known as Wi-Fi.



10. Modems

Modem is abbreviation for Modulator – De-modulator. Modems are used for data transfer from one computer network to another computer network through telephone lines. The computer network works in digital mode, while analog technology is used for carrying massages across phone lines.

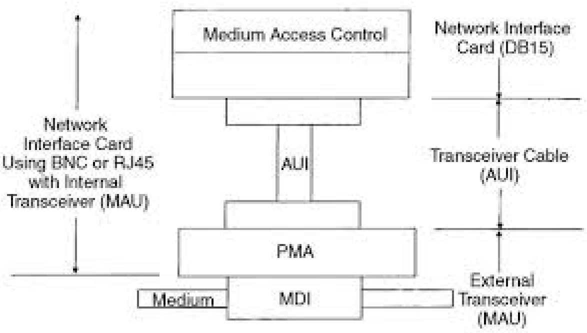
Modulator converts information from digital mode to analog mode at the transmitting end and de-modulator converts the same from analog to digital at receiving end. The process of converting analog signals of one computer network into digital signals of another computer network so they can be processed by a receiving computer is referred to as digitizing.

When an analog facility is used for data communication between two digital devices called Data Terminal Equipment (DTE), modems are used at each end. DTE can be a terminal or a computer. The modem at the transmitting end converts the digital signal generated by DTE into an analog signal by modulating a carrier. This modem at the receiving end demodulates the carrier and hand over the demodulated digital signal to the DTE.

11. Transceivers

The term transceiver does not necessarily describe a separate network device but rather an integrated technology embedded in devices such as network cards. In a network environment, a transceiver gets its name from being both a transmitter and a receiver of signals, such as analog or digital. Technically, on a LAN the transceiver is responsible to place signals onto the network media and also detecting incoming signals traveling through the same cable. Given the description of the function of a transceiver, it makes sense that that technology would be found with network cards. Although transceivers are found in network cards, they can be external devices as well. As far as networking is concerned, transceivers can ship as a module or chip type. Chip transceivers are small and are inserted into a system board or wired directly on a circuit board. Module transceivers are external to the network and are installed and function similarly to other computer peripherals, or they may function as standalone devices.

There are many types of transceivers: RF transceivers, fiber-optic transceivers, Ethernet transceivers, wireless (WAP) transceivers, and more. Though each of these media types is different, the function of the transceiver remains the same. Each type of the transceiver used has different characteristics such as the number of ports available to connect to the network and whether full-duplex communication is supported.



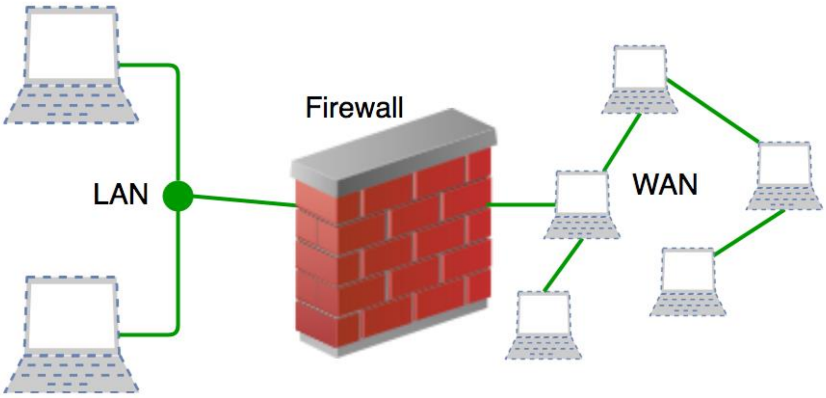
12. Firewalls

Network firewalls are security devices used to stop or mitigate unauthorized access to private networks connected to the Internet, especially intranets. The only traffic allowed on the network is defined via firewall policies – any other traffic attempting to access the network is blocked. Network firewalls sit at the front line of a network, acting as a communications liaison between internal and external devices.

A network firewall can be configured so that any data entering or exiting the network has to pass through it – it accomplishes this by examining each incoming message and rejecting those that fail to meet the defined security criteria. When properly configured, a firewall allows users to access any of the resources they need while simultaneously keeping out unwanted users, hackers, viruses, worms or other malicious programs trying to access the protected network.

Firewalls can be either hardware or software. In addition to limiting access to a protected computer and network, a firewall can log all traffic coming into or leaving a network, and manage remote access to a private network through secure authentication certificates and logins.

- ➔ Hardware firewalls: These firewalls are released either as standalone products for corporate use, or more often, as a built-in component of a router or other networking device. They are considered an essential part of any traditional security system and network configuration. Hardware firewalls will almost always come with a minimum of four network ports that allow connections to multiple systems. For larger networks, a more expansive networking firewall solution is available.
- ➔ Software firewalls: These are installed on a computer, or provided by an OS or network device manufacturer. They can be customized, and provide a smaller level of control over functions and protection features. A software firewall can protect a system from standard control and access attempts, but have trouble with more sophisticated network breaches.

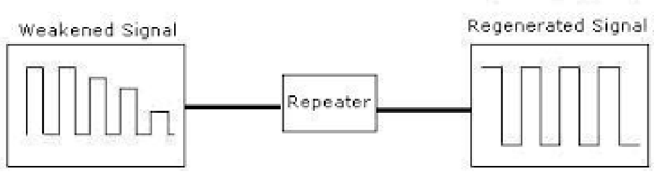


13. Repeaters

Repeaters are network devices operating at physical layer of the OSI model that amplify or regenerate an incoming signal before retransmitting it. They are incorporated in networks to expand its coverage area. They are also known as signal boosters.

When an electrical signal is transmitted via a channel, it gets attenuated depending upon the nature of the channel or the technology. This poses a limitation upon the length of the LAN or coverage area of cellular networks. This problem is alleviated by installing repeaters at certain intervals.

Repeaters amplifies the attenuated signal and then retransmits it. Digital repeaters can even reconstruct signals distorted by transmission loss. So, repeaters are popularly incorporated to connect between two LANs thus forming a large single LAN.



Back to Index

OSI MODEL IMPLEMENTATION

CODE **(Language: C++)** **(IDE: CodeBlocks)**

```
#include <iostream>

#include <cstring>

#include <ctype.h>

#include <stdlib.h>

using namespace std;

//String to character array function

void char_array(string s, char* outstr){

  int n = s.length();

  for (int i = 0; i < n; i++){

      outstr[i] = s[i];}}

//class for array functions

class ArrayFunc {

  public:

  char* message = (char*)calloc(100,sizeof(char));

  int counter;

  //Constructor to assign message the value of input

  ArrayFunc(char* outstr){

    char *temp=outstr;

    int n=0;

    while(*temp!=0) {

      n++;

      temp++;}

    counter=n;

    for (int i=0; i < n; i++){

      message[i] = outstr[i];}}
```

```
  int counters(){
    return counter;}
  void add_to_begin(char element){
    counter=counter+1;
    for(int i=counter-1;i>0;i--) {
      message[i]=message[i-1];}
    message[0]=element;}
  void add_to_end(char element){
    counter=counter+1;
    message[counter-1]=element;}
  void trim_end() {
    message[counter-1]=0;
    counter=counter-1;}
  void trim_begin() {
    for(int i=0;i<counter-1;i++) {
      message[i]=message[i+1];}
    message[counter-1]=0;
    counter=counter-1;}};
int main()
{
  char *char_arr;
  char_arr=(char*)calloc(100,sizeof(char));
  string s;
  cout<< "Enter the message to send (**without white spaces**):\n";
  cin>>s;
  //Assigning String value to the character array
  char_array(s , char_arr);
```

```
//Assigning made char array to class char array to perform functions on and keep track of array length counter.

ArrayFunc a(char_arr);

cout<<"Legend:\n7-->Application Layer\n6-->Presentation Layer\n5-->Session Layer\n4--> Transport Layer\n3-->Network Layer\n2-->Data-Link Layer\n1-->Physical Layer";

//SENDERS SIDE HEAD AND TAILS

int choice=0;

cout<<"\n\n\n-----------------------Sender's Side-----------------------\n\n\n";

cout<<"\n\nDo you want to add Application Layer\n(1->yes,0->no): ";

while(choice==0 || choice==1){

  cin>>choice;

  switch(choice){

    case 0:

      cout<<"Application layer not added";

      choice=-1;

      break;

    case 1:

      a.add_to_begin('7');

      cout<<a.message;

      choice=-1;

      break;

    default:

      cout<<"Enter Valid Value\n(1->yes,0->no): ";

      choice=0;

      break;}}

choice=0;

cout<<"\n\nDo you want to add Presentation Layer\n(1->yes,0->no): ";

while(choice==0 || choice==1){
```

```
    cin>>choice;
    switch(choice){
      case 0:
        cout<<"Presentation layer not added";
        choice=-1;
        break;
      case 1:
        a.add_to_begin('6');
        cout<<a.message;
        choice=-1;
        break;
      default:
        cout<<"Enter Valid Value\n(1->yes,0->no): ";
        choice=0;
        break;}}
  choice=0;
  cout<<"\n\nDo you want to add Session Layer\n(1->yes,0->no): ";
  while(choice==0 || choice==1){
    cin>>choice;
    switch(choice){
      case 0:
        cout<<"Session layer not added";
        choice=-1;
        break;
      case 1:
        a.add_to_begin('5');
        cout<<a.message;
```

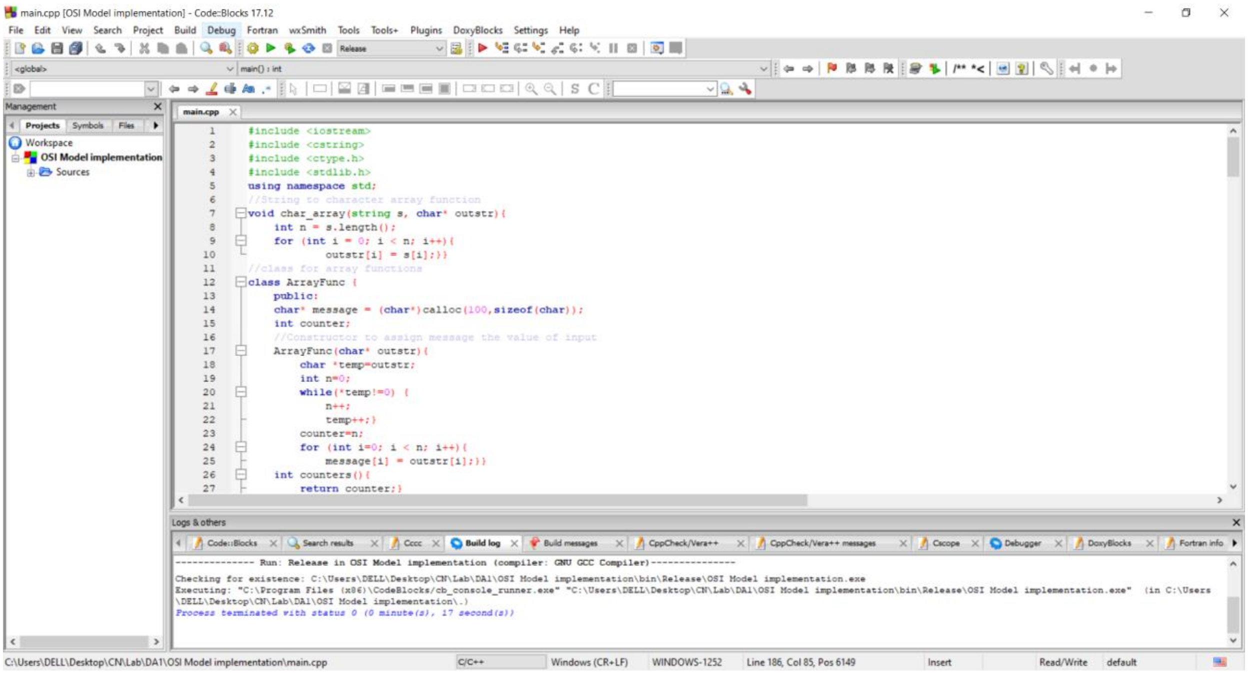
```
            choice=-1;
            break;
        default:
            cout<<"Enter Valid Value\n(1->yes,0->no): ";
            choice=0;
            break;}}
choice=0;
cout<<"\n\nDo you want to add Transport Layer\n(1->yes,0->no): ";
while(choice==0 || choice==1){
    cin>>choice;
    switch(choice){
        case 0:
            cout<<"Transport layer not added";
            choice=-1;
            break;
        case 1:
            a.add_to_begin('4');
            cout<<a.message;
            choice=-1;
            break;
        default:
            cout<<"Enter Valid Value\n(1->yes,0->no): ";
            choice=0;
            break;}}
choice=0;
cout<<"\n\nDo you want to add Network Layer\n(1->yes,0->no): ";
while(choice==0 || choice==1){
```

```
    cin>>choice;
    switch(choice){
      case 0:
        cout<<"Network layer not added";
        choice=-1;
        break;
      case 1:
        a.add_to_begin('3');
        cout<<a.message;
        choice=-1;
        break;
      default:
        cout<<"Enter Valid Value\n(1->yes,0->no): ";
        choice=0;
        break;}}
  choice=0;
  cout<<"\n\nDo you want to add Data-Link Layer\n(1->yes,0->no): ";
  while(choice==0 || choice==1){
    cin>>choice;
    switch(choice){
      case 0:
        cout<<"Data-Link layer not added";
        choice=-1;
        break;
      case 1:
        a.add_to_begin('2');
        a.add_to_end('2');
```

```
            cout<<a.message;
            choice=-1;
            break;
        default:
            cout<<"Enter Valid Value\n(1->yes,0->no): ";
            choice=0;
            break;}}
    choice=0;
    cout<<"\n\nDo you want to add Physical Layer\n(1->yes,0->no): ";
    while(choice==0 || choice==1){
        cin>>choice;
        switch(choice){
        case 0:
            cout<<"Physical layer not added";
            choice=-1;
            break;
        case 1:
            a.add_to_begin('1');
            cout<<a.message;
            choice=-1;
            break;
        default:
            cout<<"Enter Valid Value\n(1->yes,0->no): ";
            choice=0;
            break;}}
    choice=0;
    cout<<"\n\n\n-----------------------Receiver's Side-----------------------\n\n\n";
```

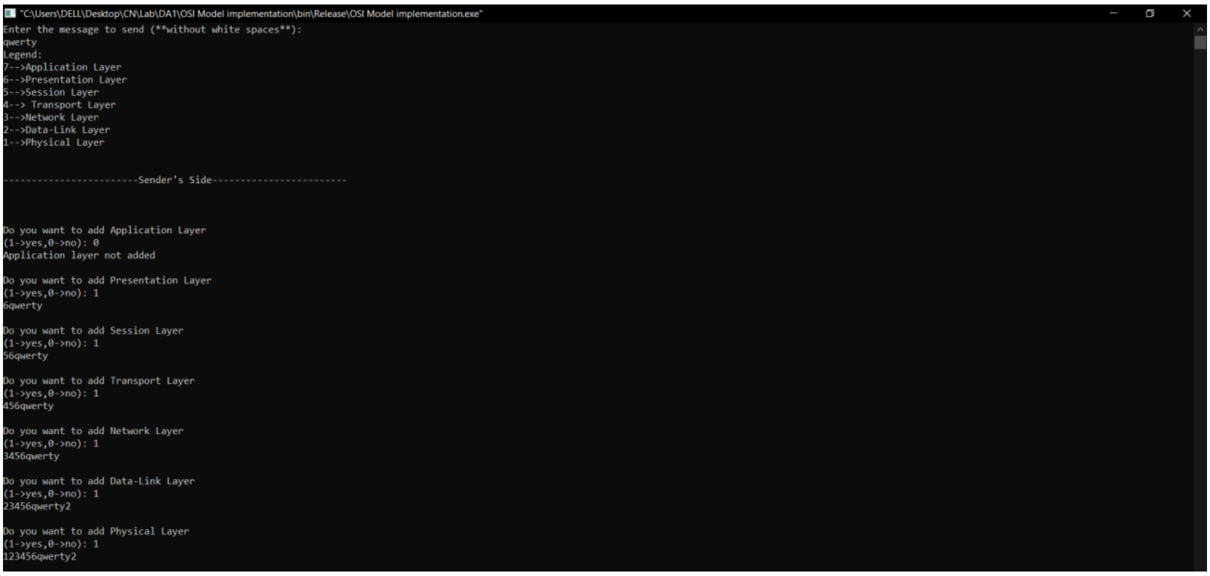
```
cout<<"Message Received: "<<a.message<<"\n\n";

while(a.message[0]=='1' || a.message[0]=='2' || a.message[0]=='3' || a.message[0]=='4' ||
a.message[0]=='5' || a.message[0]=='6' || a.message[0]=='7')

switch(a.message[0]) {

    case '1':

        a.trim_begin();

        cout<<"Physical Layer found and processed.\n"<<"Remaining encoded message is:
"<<a.message<<"\n\n";

        break;

    case '2':

        a.trim_begin();

        a.trim_end();

        cout<<"Data-Link Layer found and processed.\n"<<"Remaining encoded message is:
"<<a.message<<"\n\n";

        break;

    case '3':

        a.trim_begin();

        cout<<"Network Layer found and processed.\n"<<"Remaining encoded message is:
"<<a.message<<"\n\n";

        break;

    case '4':

        a.trim_begin();

        cout<<"Transport Layer found and processed.\n"<<"Remaining encoded message is:
"<<a.message<<"\n\n";

        break;

    case '5':

        a.trim_begin();

        cout<<"Session Layer found and processed.\n"<<"Remaining encoded message is:
"<<a.message<<"\n\n";

        break;
```

```
        case '6':
            a.trim_begin();
            cout<<"Presentation Layer found and processed.\n"<<"Remaining encoded message is: "<<a.message<<"\n\n";
            break;
        case '7':
            a.trim_begin();
            cout<<"Application Layer found and processed.\n"<<"Remaining encoded message is: "<<a.message<<"\n\n";
            break;
        default:
            cout<<"Error encountered";
            break;}
    cout<<"\n\n\nThe Received message is: "<<a.message;
    return 0;}
```

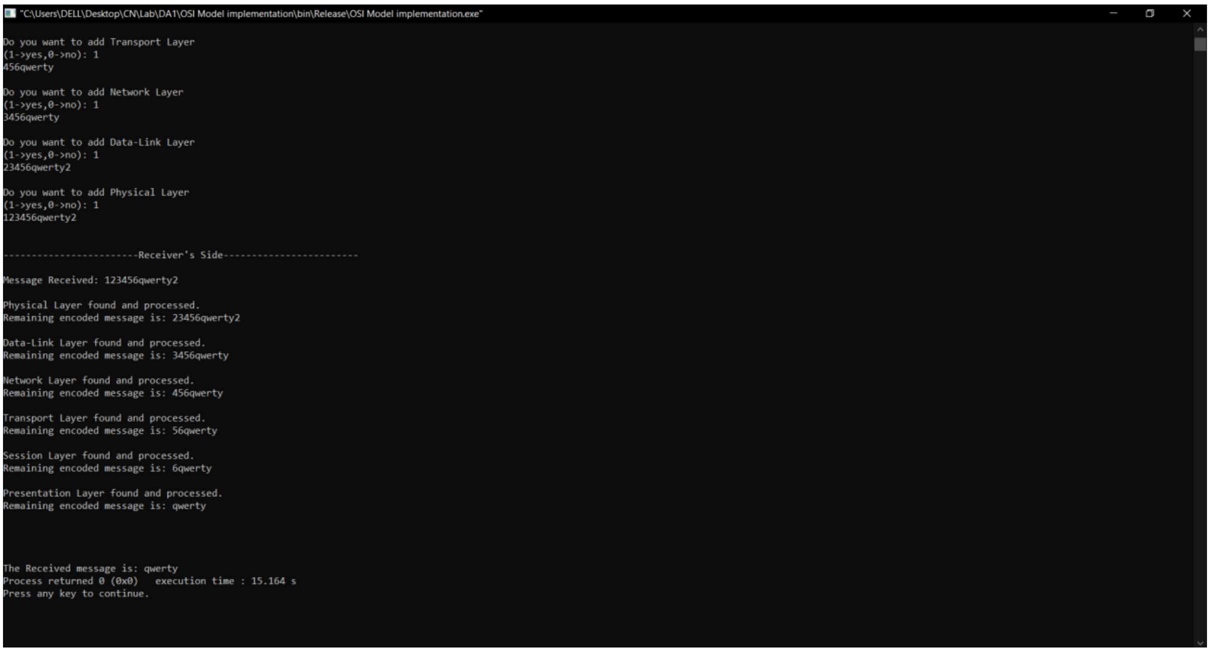


Back to Index

OUTPUT



```
"C:\Users\DELL\Desktop\CN\Lab\DA1\OSI Model implementation\bin\Release\OSI Model implementation.exe"
Enter the message to send (**without white spaces**):
qwerty
Legend:
7-->Application Layer
6-->Presentation Layer
5-->Session Layer
4--> Transport Layer
3-->Network Layer
2-->Data-Link Layer
1-->Physical Layer


-----------------------Sender's Side-----------------------



Do you want to add Application Layer
(1->yes,0->no): 0
Application layer not added

Do you want to add Presentation Layer
(1->yes,0->no): 1
6qwerty

Do you want to add Session Layer
(1->yes,0->no): 1
56qwerty

Do you want to add Transport Layer
(1->yes,0->no): 1
456qwerty

Do you want to add Network Layer
(1->yes,0->no): 1
3456qwerty

Do you want to add Data-Link Layer
(1->yes,0->no): 1
23456qwerty2

Do you want to add Physical Layer
(1->yes,0->no): 1
123456qwerty2
```



```
"C:\Users\DELL\Desktop\CN\Lab\DA1\OSI Model implementation\bin\Release\OSI Model implementation.exe"

Do you want to add Transport Layer
(1->yes,0->no): 1
456qwerty

Do you want to add Network Layer
(1->yes,0->no): 1
3456qwerty

Do you want to add Data-Link Layer
(1->yes,0->no): 1
23456qwerty2

Do you want to add Physical Layer
(1->yes,0->no): 1
123456qwerty2


-----------------------Receiver's Side-----------------------

Message Received: 123456qwerty2

Physical Layer found and processed.
Remaining encoded message is: 23456qwerty2

Data-Link Layer found and processed.
Remaining encoded message is: 3456qwerty

Network Layer found and processed.
Remaining encoded message is: 456qwerty

Transport Layer found and processed.
Remaining encoded message is: 56qwerty

Session Layer found and processed.
Remaining encoded message is: 6qwerty

Presentation Layer found and processed.
Remaining encoded message is: qwerty



The Received message is: qwerty
Process returned 0 (0x0)   execution time : 15.164 s
Press any key to continue.
```


Back to Index