Experiment: 01

Objective: Computer Network Topologies Simulation tool using CISCO Packet Tracer.

Introduction:

Topology

• Topology refers to the layout of connected devices on a network. Here, some logical layout of topology.

Mesh Topology

• Here every device has a point to point link to every other device.
• Node 1 node must be connected with n-1 nodes.
• A fully connected mesh can have n(n-1)/2 physical channels to link n devices.
• It must have n-1 I/O ports.

Advantages:

1. They use dedicated links so each link can only carry its own data load. So traffic problem can be avoided.
2. It is robust. If any one link get damaged it cannot affect others.
3. It gives privacy and security.(Message travels along a dedicated link)
4. Fault identification and fault isolation are easy.

Disadvantages:

1. The amount of cabling and the number of I/O ports required are very large. Since every device is connected to each devices through dedicated links.
2. The sheer bulk of wiring is larger then the available space.
3. Hardware required to connected each device is highly expensive.

Applications:

1. Telephone Regional office.
2. WAN.(Wide Area Network).

Star Topology

• Here each device has a dedicated point-to-point link to the central controller called "Hub"(Act as a Exchange).
• There is no direct traffic between devices.
• The transmission are occurred only through the central "hub".
• When device 1 wants to send data to device 2; First sends the data to hub. Which then relays the data to the other connected device.

Advantages:

1. Less expensive then mesh since each device is connected only to the hub.
2. Installation and configuration are easy.
3. Less cabling is need then mesh.
4. Robustness.(if one link fails, only that links is affected. All other links remain active)
5. Easy to fault identification & to remove parts.
6. No distruptions to the network then connecting(or) removing devices.

Disadvantages:

1. Even it requires less cabling then mesh when compared with other topologies it still large.(Ring or bus).
2. Dependency(whole n/w dependent on one single point(hub). When it goes down. The whole system is dead.

Applications

- Star topology used in Local Area Networks(LANs).
- High speed LAN often used STAR.

Bus Topology

- A bus topology is multipoint.
- Here one long cable act as a backbone to link all the devices are connected to the backbone by drop lines and taps.
- Drop line- is the connection b/w the devices and the cable.
- Tap- is the splitter that cut the main link.
- This allows only one device to transmit at a time.
- A device want to communicate with other device on the n/ws sends a broadcast message onto the wire all other devices see.
- But only the intended devices accepts and process the message.

Advantages:

1. Ease of installation
2. Less cabling

Disadvantages:

1. Difficult reconfiguration and fault isolation.
2. Difficult to add new devices.
3. Signal reflection at top can degradation in quality.
4. If any fault in backbone can stops all transmission.

Applications:

Most computer motherboard.

Ring Topology

- Here each device has a dedicated connection with two devices on either side.
- The signal is passed in one direction from device to device until it reaches the destination and each device have repeater.
- When one device received signals instead of intended another device, its repeater then regenerates the data and passes them along.
- To add or delete a device requires changing only two connections.

Advantages:

1. Easy to install.
2. Easy to reconfigure.
3. Fault identification is easy.

Disadvantages:

1. Unidirectional traffic.
2. Break in a single ring can break entire network.

Applications:

- Ring topologies are found in some office buildings or school campuses.
- Today high speed LANs made this topology less popular.

Tree Topology

- Alternatively referred to as a star bus topology.
- Tree topology is one of the most common network setups that is similar to a bus topology and a star topology.
- A tree topology connects multiple star networks to other star networks. Below is a visual example of a simple computer setup on a network using the star topology.

Hybrid Topology

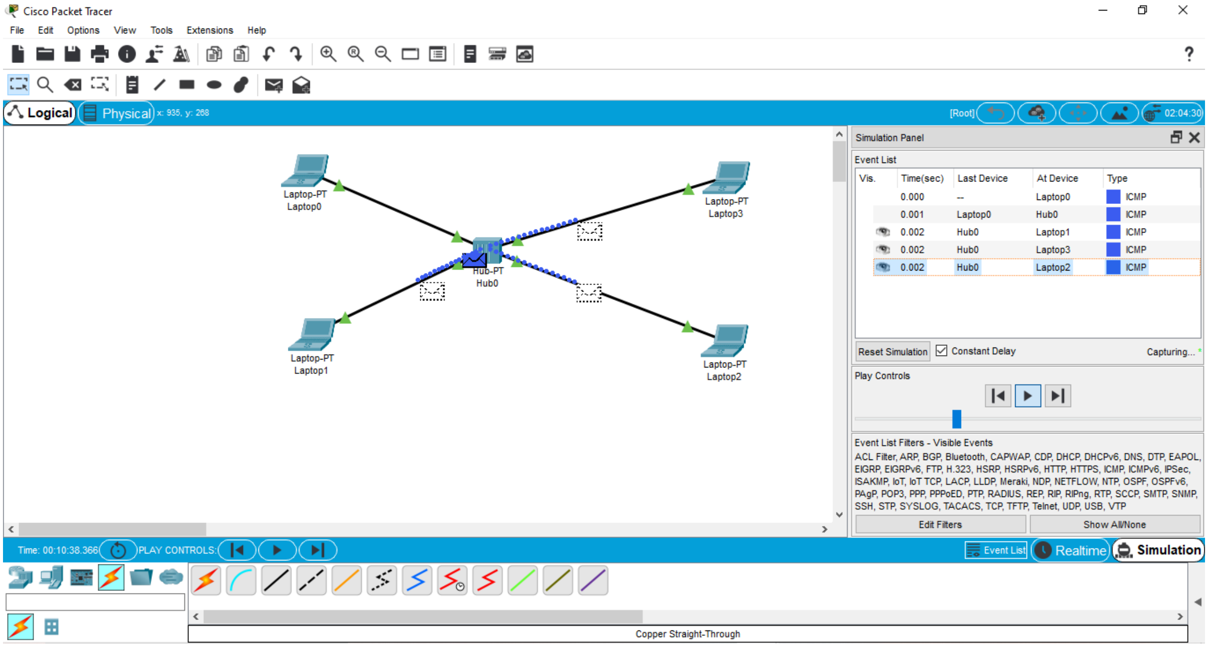
- A network which contain all type of physical structure and connected under a single backbone channel.

Considerations for choosing topology

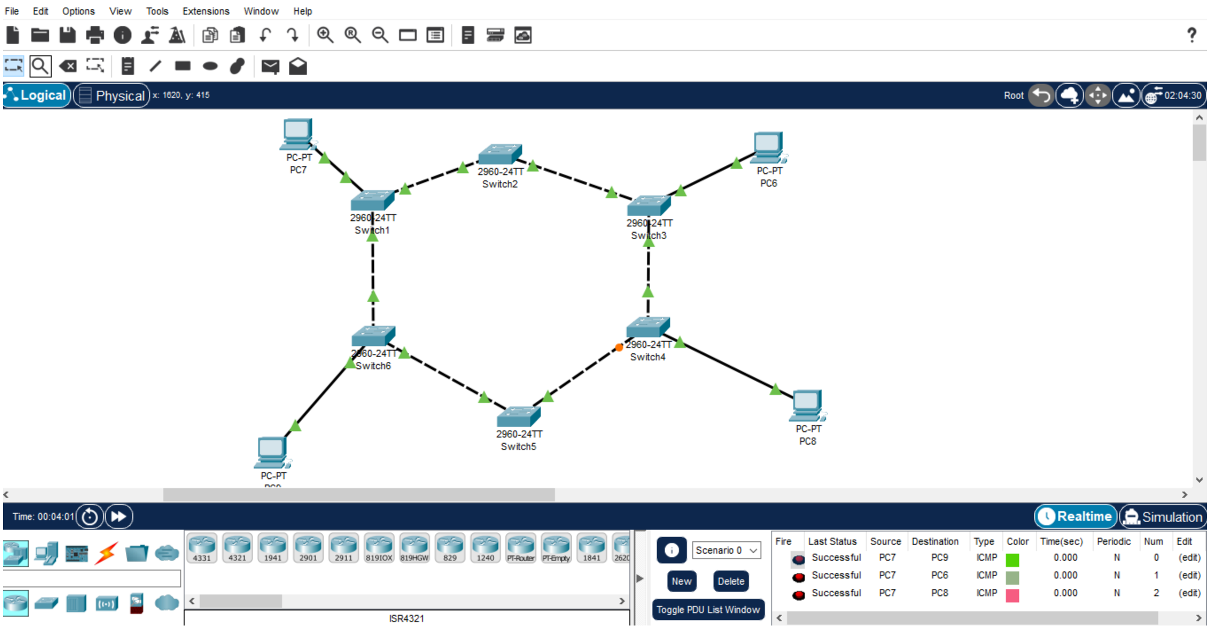
- Money-Bus n/w may be the least expensive way to install a n/w.
- Length-of cable needed- the linear bus n/w uses shorter lengths of cable.
- Future growth-with star topology, expending a n/w is easily done by adding another devices.
- Cable type-most common used cable in commercial organization is twisted pair. Which often used with star topologies.
- Full mesh topology is theoretically the best since every device is connected to every other device.(thus maximizing speed and security. however, it quite expensive to install)
- Next best would be tree topology, which is basically a connection of star.

Outputs

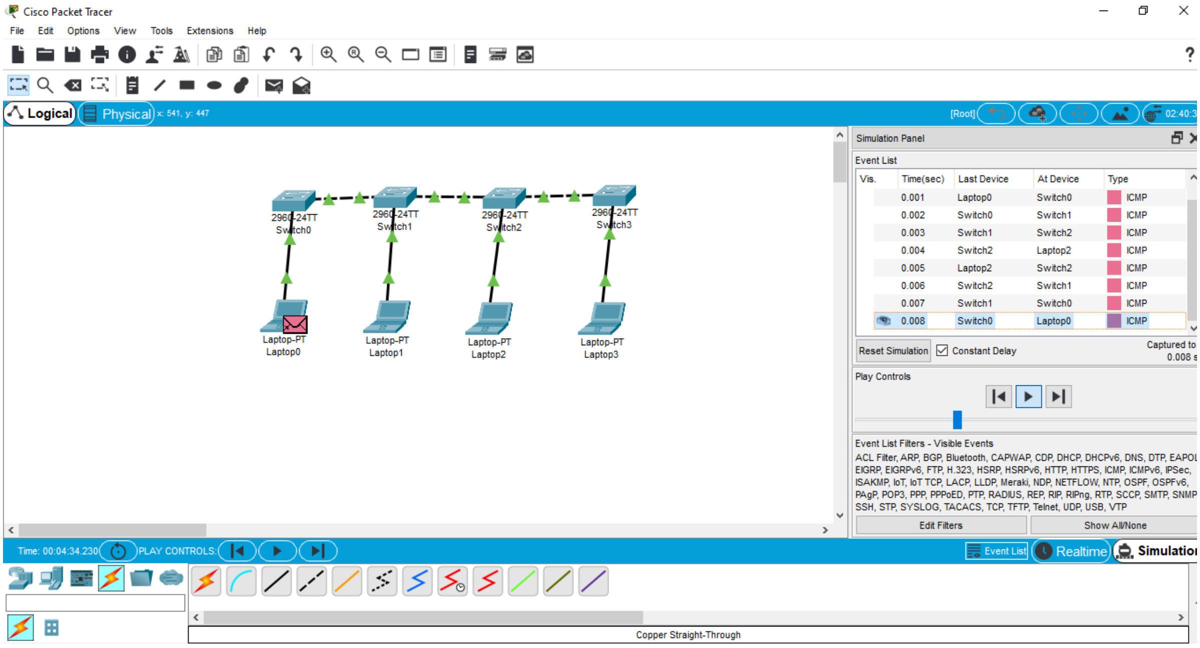
Star Topology

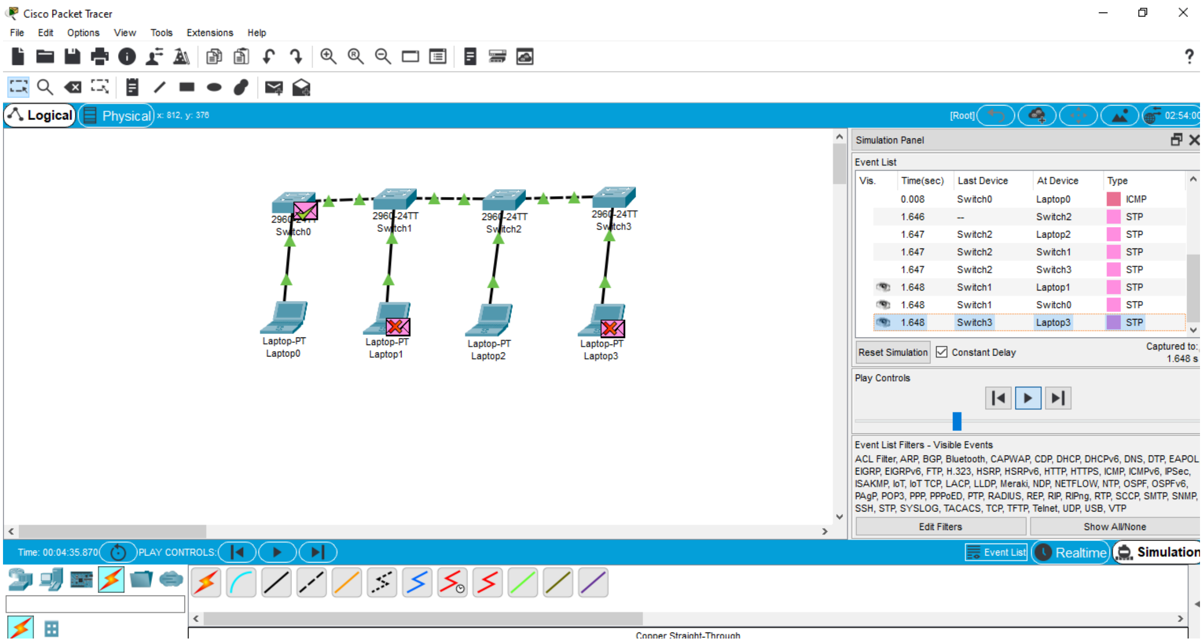


Ring Topology

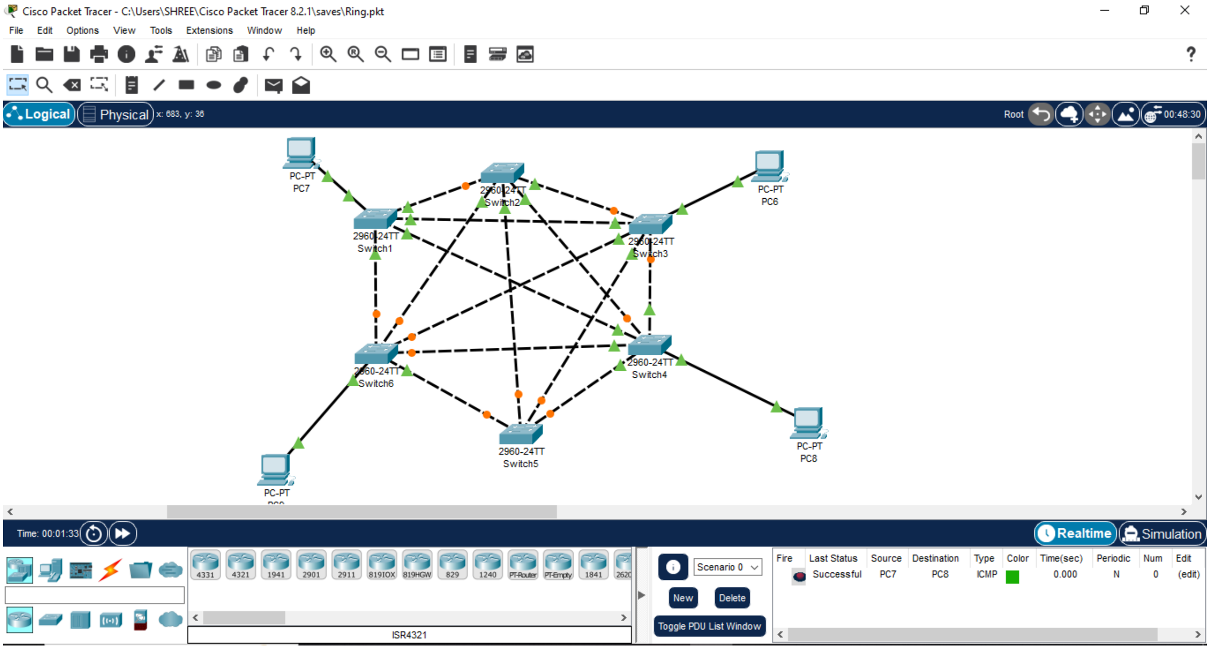


Bus Topology

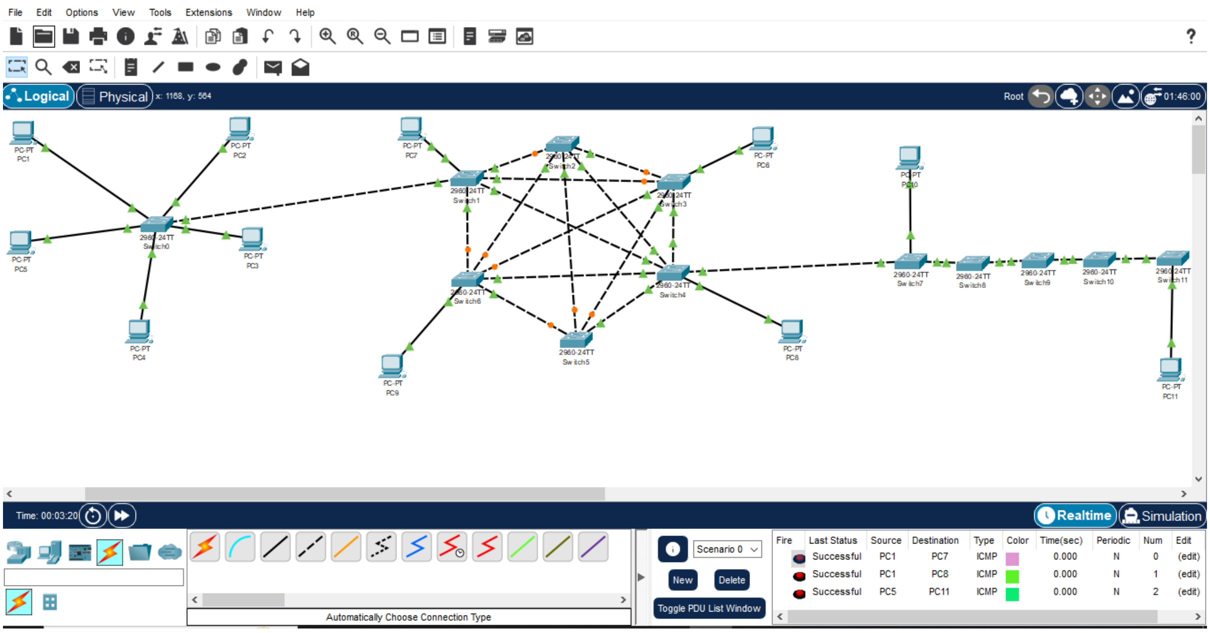




Mesh Topology



Hybrid Topology



Q. Which topologies are used in the College Computer Laboratories? Why?

Experiment: 02

Objective: CLI commands for switch-to-switch configuration

Introduction:

CLI stands for Command Line Interface. It is a user interface that allows users to interact with network devices through commands using only the keyboard. There are many network engineers who prefer the command line interface because of its faster processing speed than the graphical user interface. In the networking world, not all problems can be solved through a graphical user interface. Therefore, engineers must use the command line interface to solve problems, not the graphical user interface.

- It does not require pointing devices like a Mouse for selecting and choosing items. It only needs a keyboard.
- In the command line interface, spelling and typo errors cannot be avoided.
- The command-line interface is harder to use than the graphical user interface.
- The command-line interface requires less memory.

CLI advantages and disadvantages

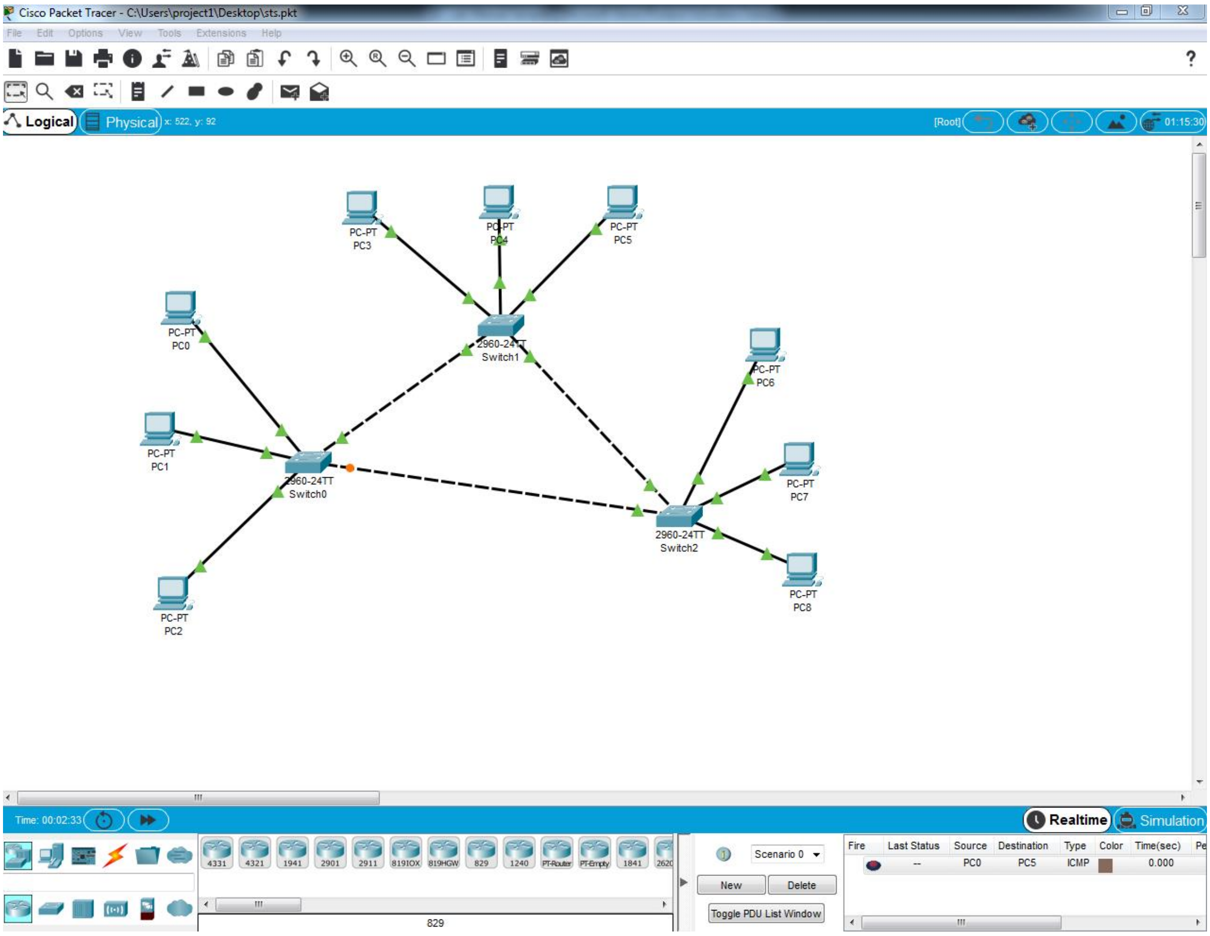
The following are advantages of a command-line interface:

- granular control of an OS or application;
- more efficient management of a large number of systems;
- ability to store scripts to automate regular tasks; and
- basic command-line interface knowledge can enable troubleshooting of network connection issues or resolving other system tasks.

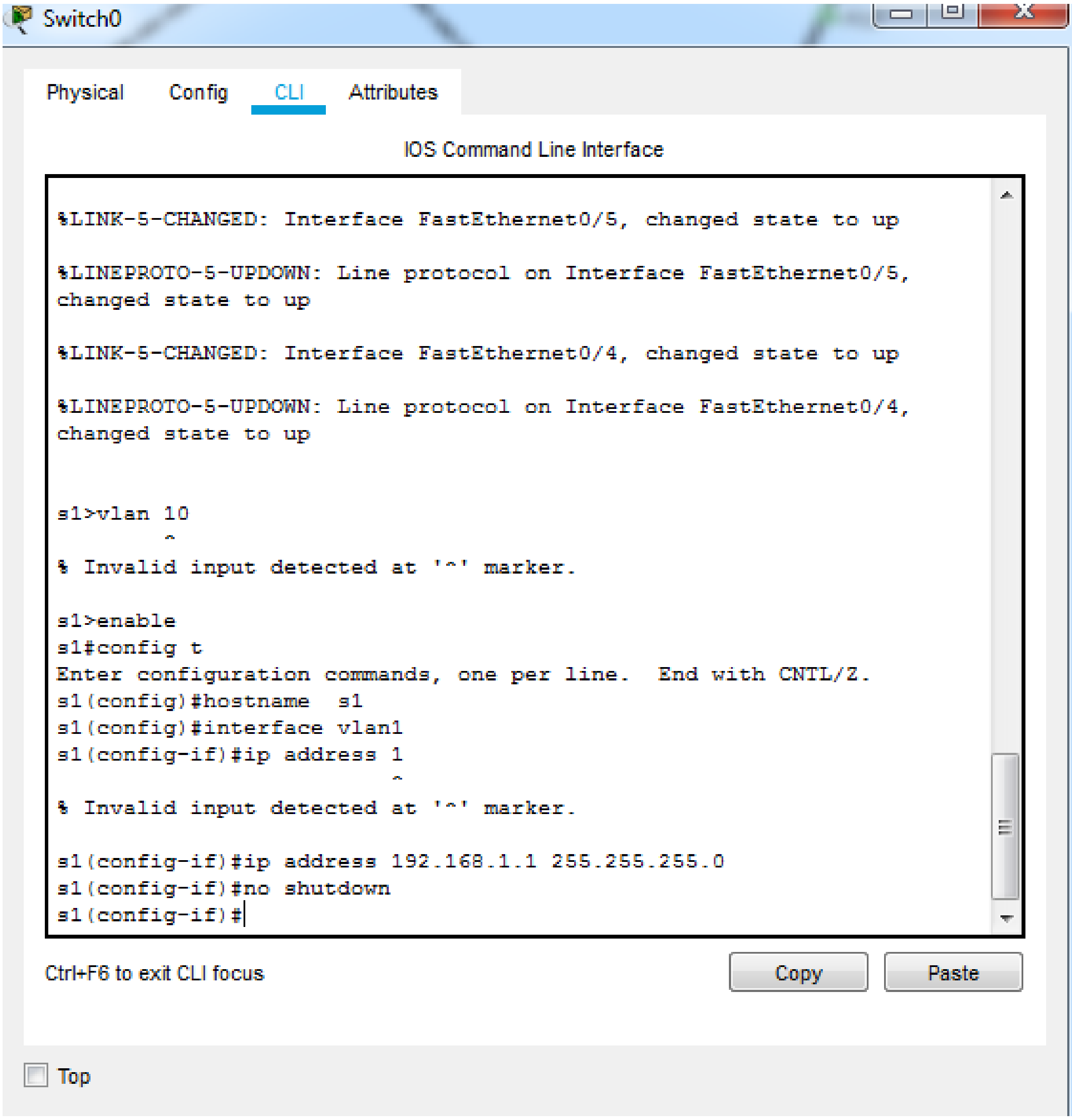
The disadvantages of a command-line interface are the following:

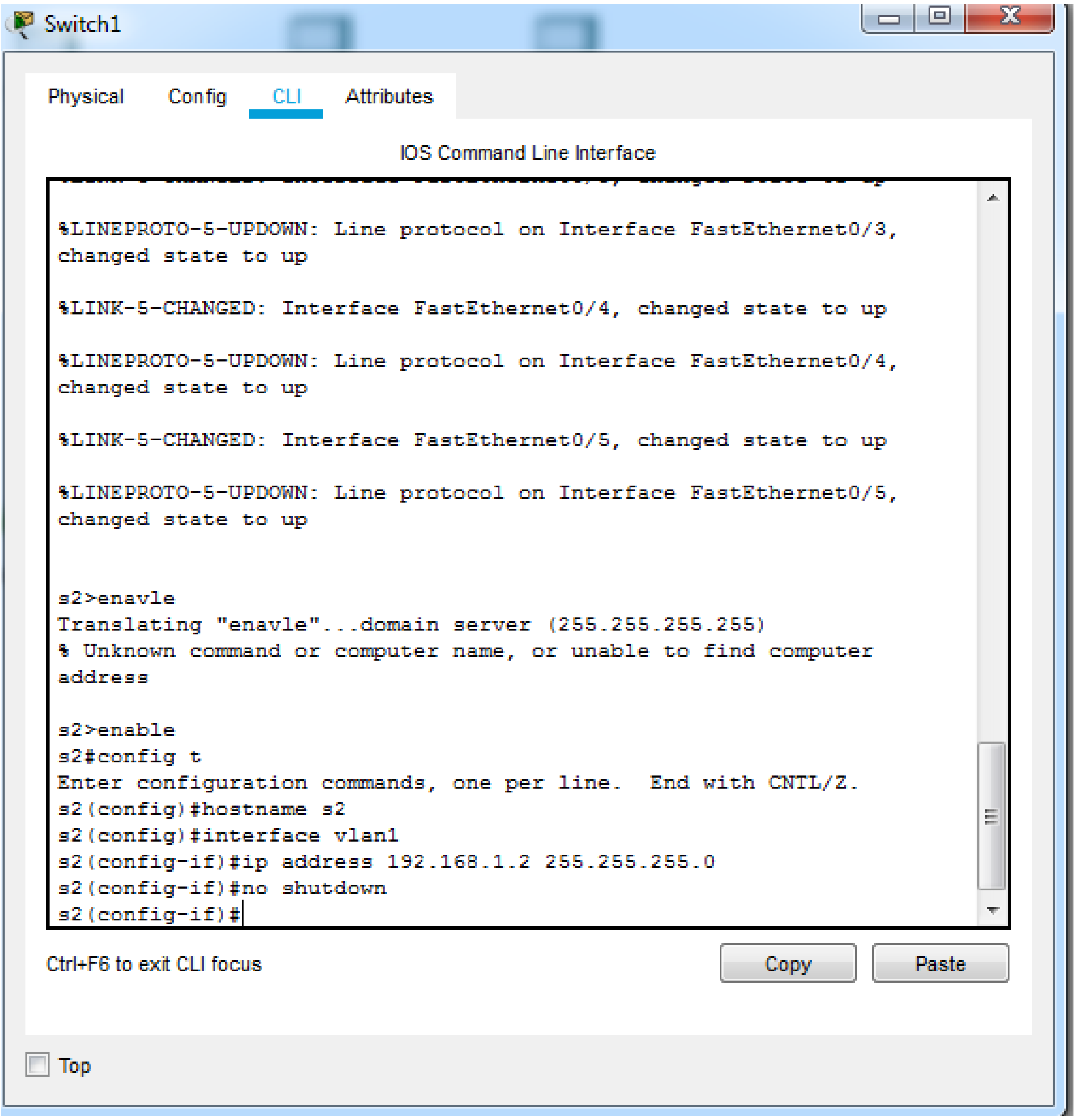
- GUI is more user friendly;
- steeper learning curve associated with memorizing commands and complex syntax/arguments; and
- different commands used in different shells.

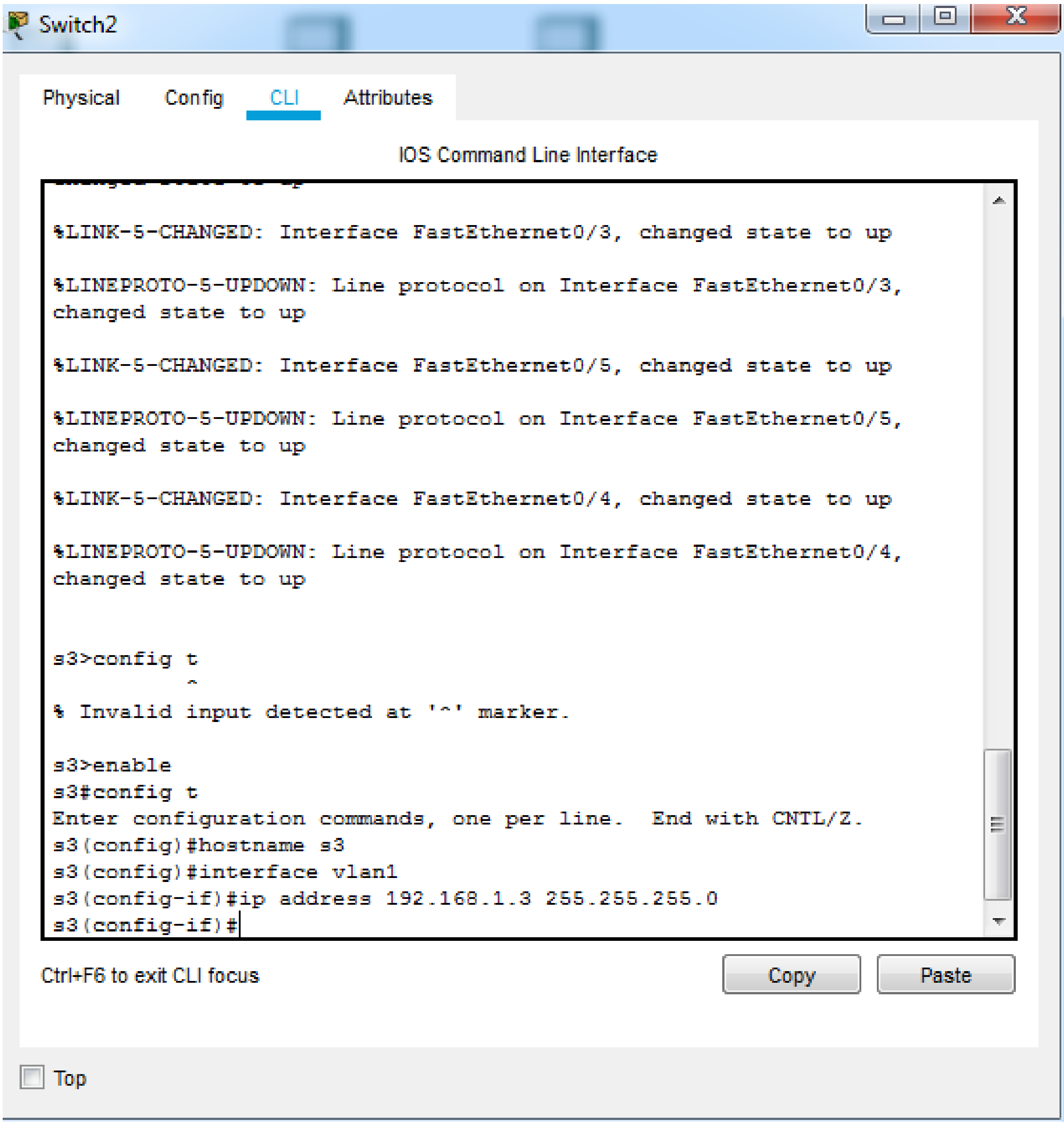
Conclusion: CLI commands for switch to switch configuration has been verified practically on cisco packet tracker.

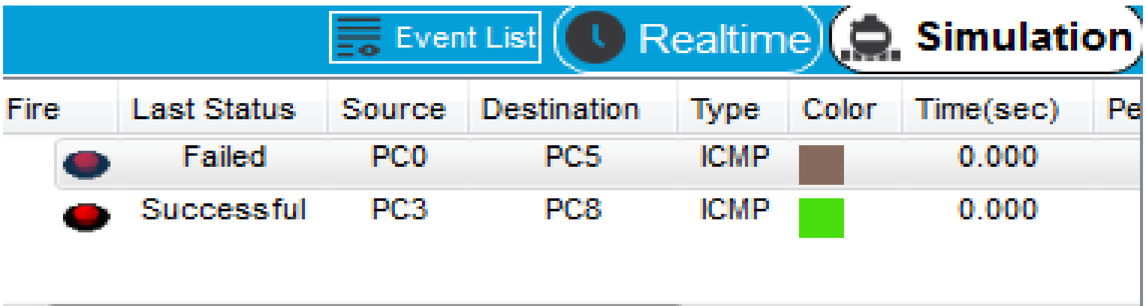


Cisco Packet Tracer - C:\Users\project1\Desktop\sts.pkt
Logical
Physical
PC-PT
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
2960-24TT
Switch0
Switch1
Switch2
Time: 00:02:33
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
PC0
PC5
ICMP
0.000


Switch0
Physical
Config
CLI
Attributes
IOS Command Line Interface
%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/4, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed state to up
s1>vlan 10
^
% Invalid input detected at '^' marker.
s1>enable
s1#config t
Enter configuration commands, one per line. End with CNTL/Z.
s1(config)#hostname s1
s1(config)#interface vlan1
s1(config-if)#ip address 1
^
% Invalid input detected at '^' marker.
s1(config-if)#ip address 192.168.1.1 255.255.255.0
s1(config-if)#no shutdown
s1(config-if)#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top


Switch1
Physical
Config
CLI
Attributes
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/4, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up
s2>enavle
Translating "enavle"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
s2>enable
s2#config t
Enter configuration commands, one per line. End with CNTL/Z.
s2(config)#hostname s2
s2(config)#interface vlan1
s2(config-if)#ip address 192.168.1.2 255.255.255.0
s2(config-if)#no shutdown
s2(config-if)#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top


Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
%LINK-5-CHANGED: Interface FastEthernet0/3, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/4, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed state to up
s3>config t
^
% Invalid input detected at '^' marker.
s3>enable
s3#config t
Enter configuration commands, one per line. End with CNTL/Z.
s3(config)#hostname s3
s3(config)#interface vlan1
s3(config-if)#ip address 192.168.1.3 255.255.255.0
s3(config-if)#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top


Event List
Realtime
Simulation
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Pe
Failed
PC0
PC5
ICMP
0.000
Successful
PC3
PC8
ICMP
0.000

Experiment 3

Objective: To build a Virtual LAN (VLAN) between PC and server machines , and to establish VLAN connections to avoid the interference between these two.

Introduction: A virtual local area network (VLAN) is a virtualized connection that connects multiple devices and network nodes from different LANs into one logical network.

Virtual local area networks have become crucial for organizations with complex networking systems. Organizations require solutions that allow them to scale their networks, segment them to increase security measures, and decrease network latency. While LAN is used to connect a group of devices such as computers and printers to a server via cables, VLANs allow multiple LANs and associated devices to communicate via wireless internet. Outlined below are the five different types of virtual networks:

- Management virtual local area network.
- Voice virtual local area network.
- Native virtual local area network.
- Default virtual local area network.
- Data virtual local area network.

Advantages:

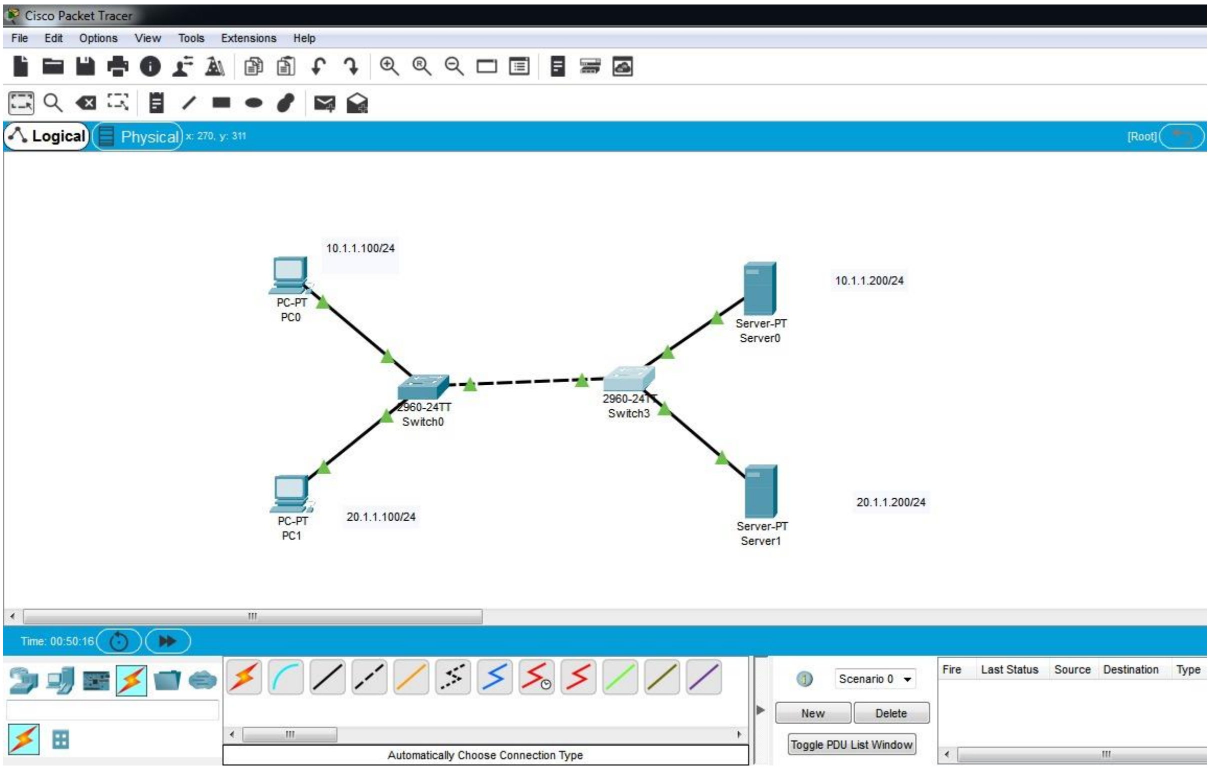
1. It solves a broadcast problem.
2. VLAN reduces the size of broadcast domains.
3. VLAN allows you to add an additional layer of security.
4. It can make device management simple and easier.
5. You can make a logical grouping of devices by function rather than location.
6. It allows you to create groups of logically connected devices that act like they are on their own network.

Disadvantages:

1. A packet can leak from one VLAN to other.
2. An injected packet may lead to a cyber-attack.
3. Threat in a single system may spread a virus through a whole logical network.
4. You require an additional router to control the workload in large networks.
5. You can face problems in interoperability.
6. A VLAN cannot forward network traffic to other VLANs.

Conclusion: Virtual LAN (VLAN) using cisco packet tracker has been verified practically.

Output: Network topology implementation using CISCO Packet tracer



Physical Config CLI Attributes

IOS Command Line Interface

```
Switch>en
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname sw0
sw0(config)#exit
sw0#
%SYS-5-CONFIG_I: Configured from console by console

sw0#sh vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
sw0#
sw0#config t
Enter configuration commands, one per line.  End with CNTL/Z.
sw0(config)#vlan 10
sw0(config-vlan)#name Accounts
sw0(config-vlan)#
sw0(config-vlan)#vlan 20
sw0(config-vlan)#name HR
sw0(config-vlan)#exit
sw0(config)#exit
sw0#
%SYS-5-CONFIG_I: Configured from console by console

sw0#sh vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
10   Accounts                         active
20   HR                               active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
sw0#
sw0#config t
```

Physical Config CLI Attributes

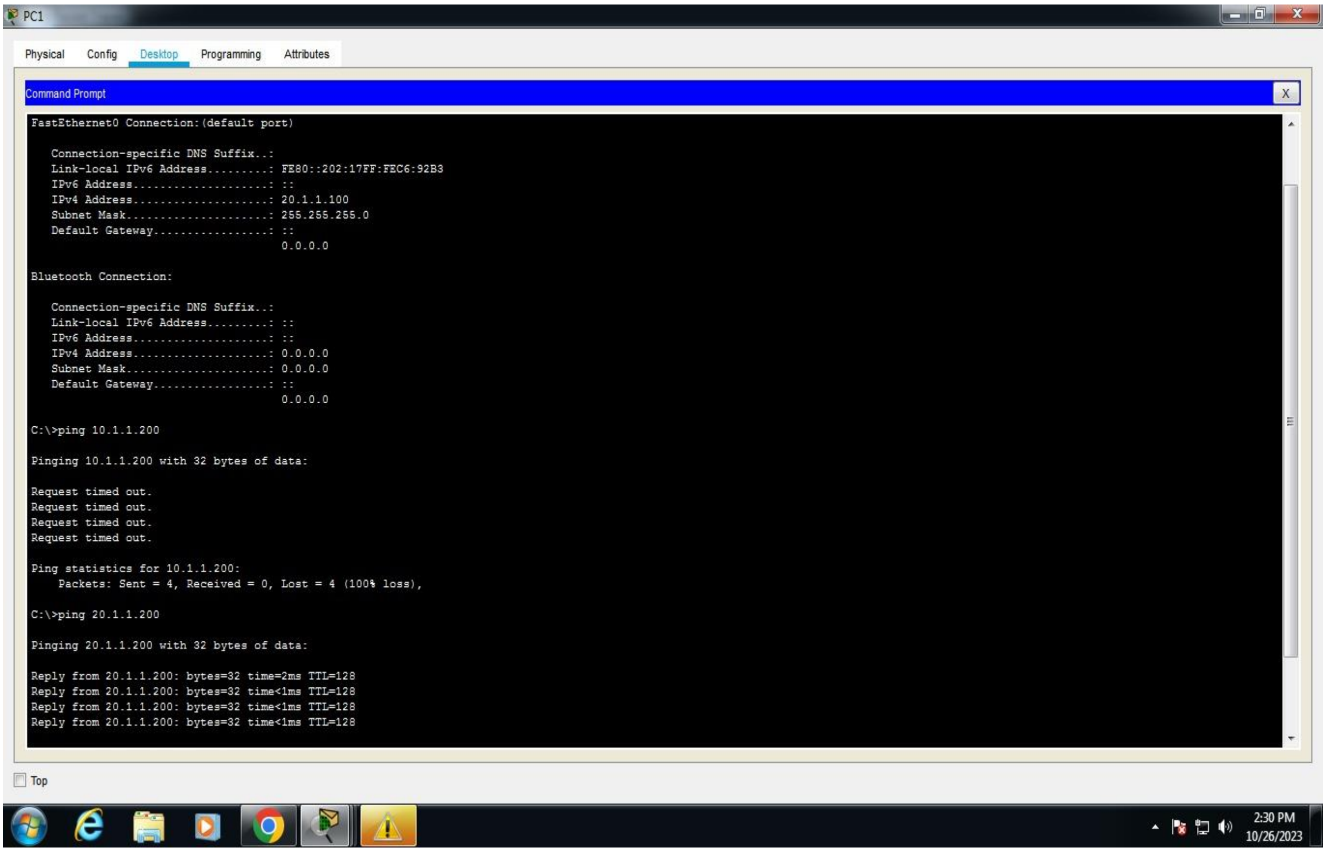
IOS Command Line

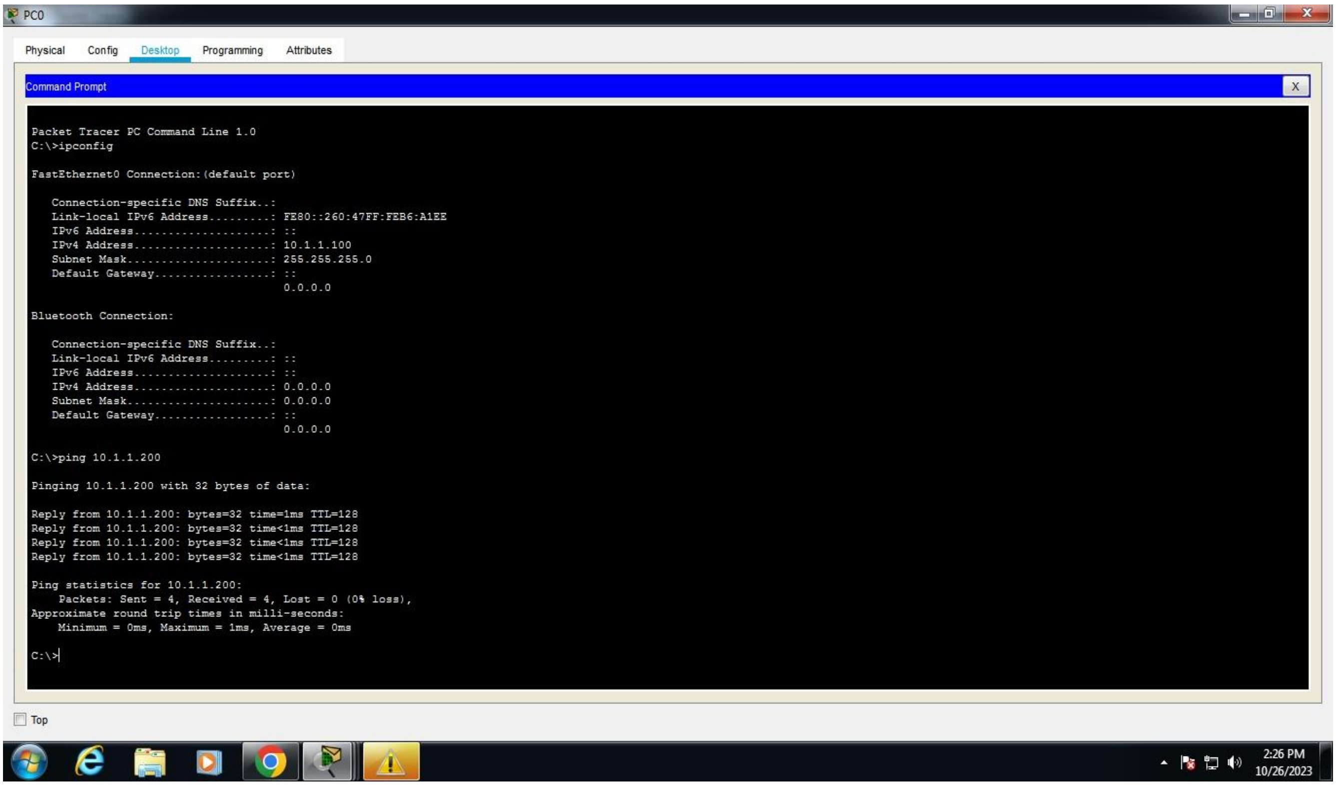
```
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
sw0#
sw0#config t
Enter configuration commands, one per line.  End with CNTL/Z.
sw0(config)#int fa0/2
sw0(config-if)#switchport mode access
sw0(config-if)#switchport access vlan 10
sw0(config-if)#
sw0(config-if)#
sw0(config-if)#int fa0/3
sw0(config-if)#switchport mode access
sw0(config-if)#switchport access vlan 20
sw0(config-if)#exit
sw0(config)#exit
sw0#
%SYS-5-CONFIG_I: Configured from console by console

sw0#sh vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/4, Fa0/5, Fa0/6
                                                Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24, Gig0/1, Gig0/2
10   Accounts                         active    Fa0/2
20   HR                               active    Fa0/3
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
sw0#
sw0#
sw0#int fa0/1
        ^
% Invalid input detected at '^' marker.

sw0#config t
Enter configuration commands, one per line.  End with CNTL/Z.
sw0(config)#int fa0/1
sw0(config-if)#switchport mode trunk
sw0(config-if)#end
sw0#
%SYS-5-CONFIG_I: Configured from console by console
```


PC1
Physical
Config
Desktop
Programming
Attributes
Command Prompt
FastEthernet0 Connection:(default port)
Connection-specific DNS Suffix..:
Link-local IPv6 Address.........: FE80::202:17FF:FEC6:92B3
IPv6 Address....................: ::
IPv4 Address....................: 20.1.1.100
Subnet Mask.....................: 255.255.255.0
Default Gateway.................: ::
0.0.0.0
Bluetooth Connection:
Connection-specific DNS Suffix..:
Link-local IPv6 Address.........: ::
IPv6 Address....................: ::
IPv4 Address....................: 0.0.0.0
Subnet Mask.....................: 0.0.0.0
Default Gateway.................: ::
0.0.0.0
C:\>ping 10.1.1.200
Pinging 10.1.1.200 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.1.1.200:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 20.1.1.200
Pinging 20.1.1.200 with 32 bytes of data:
Reply from 20.1.1.200: bytes=32 time=2ms TTL=128
Reply from 20.1.1.200: bytes=32 time<1ms TTL=128
Reply from 20.1.1.200: bytes=32 time<1ms TTL=128
Reply from 20.1.1.200: bytes=32 time<1ms TTL=128
Top
2:30 PM
10/26/2023


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Packet Tracer PC Command Line 1.0
C:\>ipconfig
FastEthernet0 Connection:(default port)
Connection-specific DNS Suffix..:
Link-local IPv6 Address.........: FE80::260:47FF:FEB6:A1EE
IPv6 Address....................: ::
IPv4 Address....................: 10.1.1.100
Subnet Mask.....................: 255.255.255.0
Default Gateway.................: ::
0.0.0.0
Bluetooth Connection:
Connection-specific DNS Suffix..:
Link-local IPv6 Address.........: ::
IPv6 Address....................: ::
IPv4 Address....................: 0.0.0.0
Subnet Mask.....................: 0.0.0.0
Default Gateway.................: ::
0.0.0.0
C:\>ping 10.1.1.200
Pinging 10.1.1.200 with 32 bytes of data:
Reply from 10.1.1.200: bytes=32 time=1ms TTL=128
Reply from 10.1.1.200: bytes=32 time<1ms TTL=128
Reply from 10.1.1.200: bytes=32 time<1ms TTL=128
Reply from 10.1.1.200: bytes=32 time<1ms TTL=128
Ping statistics for 10.1.1.200:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>
Top
2:26 PM
10/26/2023

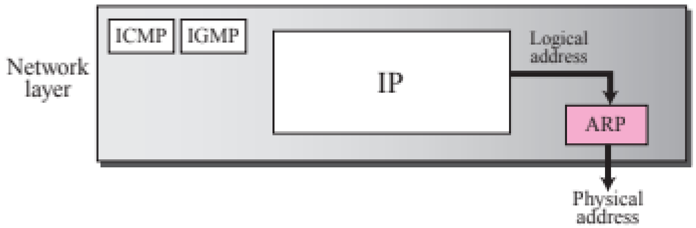
Experiment. 4

Objective: To investigate ARP protocol using Wireshark.

Introduction

Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. A mapping corresponds a logical address to a phys- ical address. Figure 8.1 shows the position of the ARP in the TCP/IP protocol suite. ARP accepts a logical address from the IP protocol, maps the address to the corresponding physical address and pass it to the data link layer.

Position of ARP in TCP/IP protocol suite:



ARP Packet :

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

❑**Hardware type.** This is a 16-bit field defining the type of the network on which

ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given the type 1. ARP can be used on any physical network.

❑ **Protocol type**. This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016. ARP can be used with any higher-level protocol.

❑ **Hardware length.** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.

❑ **Protocol length.** This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.

❑ **Operation.** This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1), ARP reply (2).

❑ **Sender hardware address**. This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.

❑ **Sender protocol address.** This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.

❑ **Target hardware address.** This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all 0s because the sender does not know the physical address of the target.

❑ **Target protocol address**. This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long

Working of Address Resolution Protocol (ARP)

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

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

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

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

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

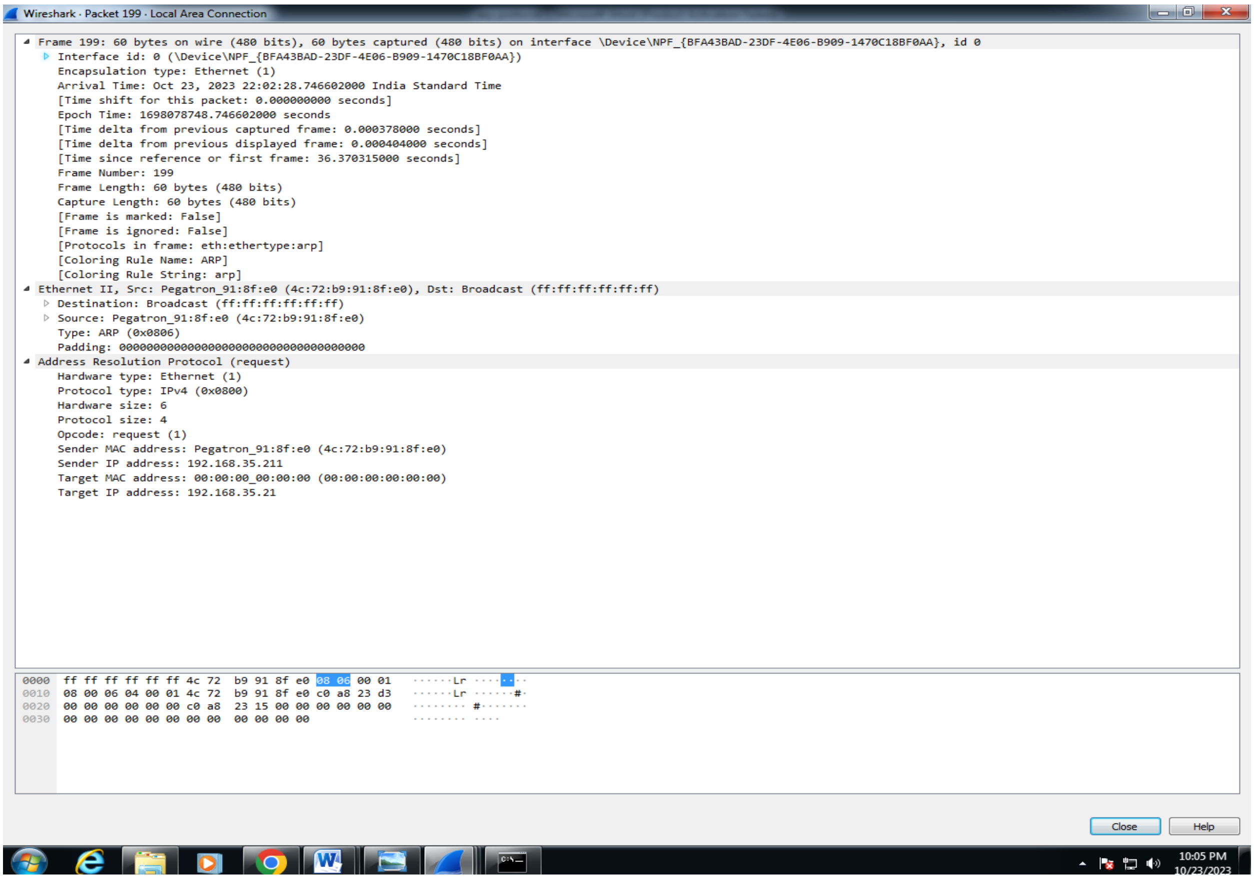
Step 6: The destination device will update its Address Resolution Protocol (ARP) cache, since it need to contact the sender machine soon.

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

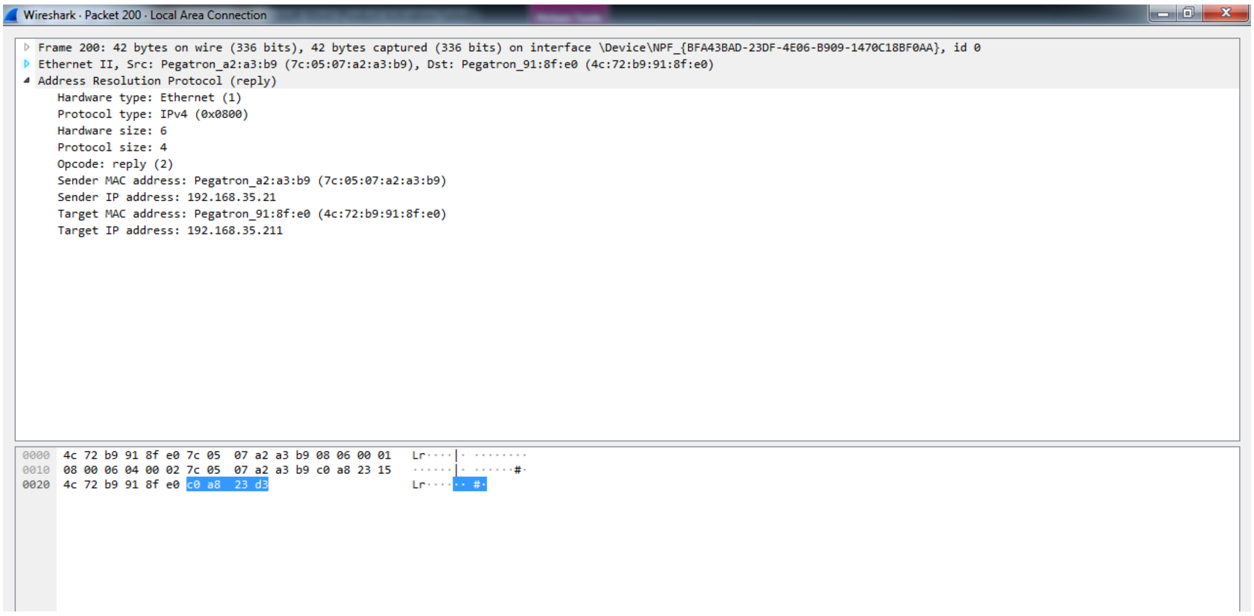
Step 8: The source machine will process the Address Resolution Protocol (ARP) reply from destination, it store the Sender Hardware Address as the layer 2 address of the destination.

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

Output: ARP Request:



ARP Reply :



Experiment No. 5

Objective: To investigate DNS protocol using Wireshark.

Introduction :

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as DNS resolver sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

- Fig 8.1 Purpose Of DNS :

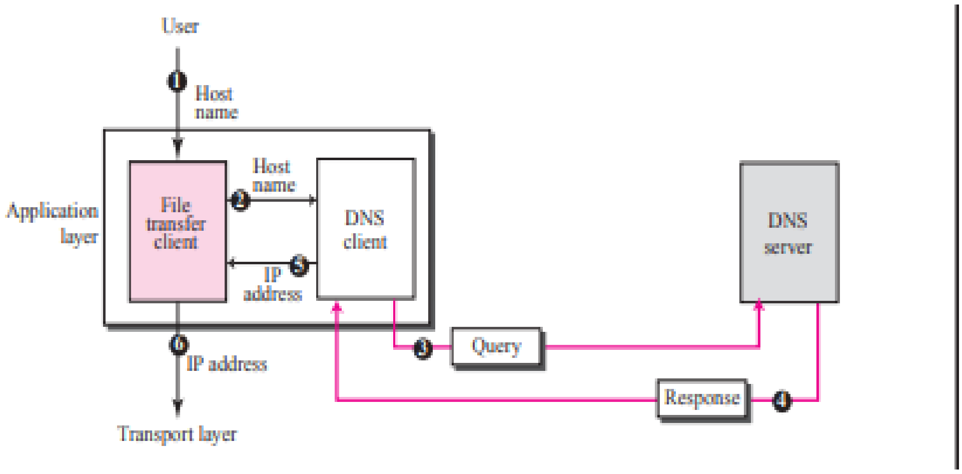


Fig 8.2 Query And Response Messeges

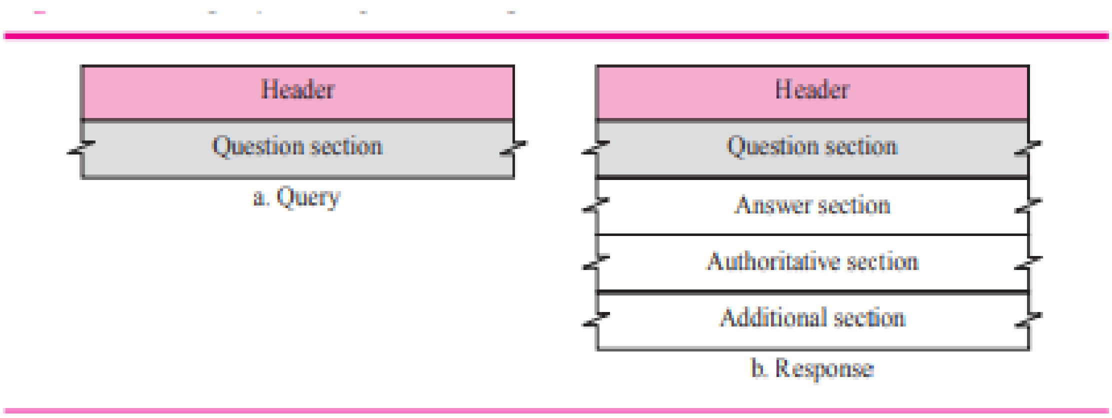


Fig 8.3 Header format

Identification	Flags
Number of question records	Number of answer records (All 0s in query message)
Number of authoritative records (All 0s in query message)	Number of additional records (All 0s in query message)

❑ Identification. This is a 16-bit field used by the client to match the response with the query. The client uses a different identification number each time it sends a query. The server duplicates this number in the corresponding response.

❑ Flags. This is a 16-bit field consisting of the subfields shown in Figure 8.3.

A brief description of each flag subfield follows. a. QR (query/response).

This is a 1-bit subfield that defines the type of message. If it is 0, the message is a query. If it is 1, the message is a response.

b. OpCode. This is a 4-bit subfield that defines the type of query or response (0 if standard, 1 if inverse, and 2 if a server status request).

c. AA (authoritative answer). This is a 1-bit subfield. When it is set (value of 1) it means that the name server is an authoritative server. It is used only in a response message.

d. TC (truncated). This is a 1-bit subfield. When it is set (value of 1), it means that the response was more than 512 bytes and truncated to 512. It is used when DNS uses the services of UDP (see Section 8.3 on Encapsulation).

e. RD (recursion desired). This is a 1-bit subfield. When it is set (value of 1) it means the client desires a recursive answer. It is set in the query message and repeated in the response message.

f. RA (recursion available). This is a 1-bit subfield. When it is set in the response, it means that a recursive response is available. It is set only in the response message

g. Reserved. This is a 3-bit subfield set to 000.

h. rCode. This is a 4-bit field that shows the status of the error in the response. Of course, only an authoritative server can make such a judgment. Table 8.2 shows the possible values for this field.

Output (Query) :

dns

No.	Time	Source	Destination	Protocol	Length	Info
30342	56.632426	192.168.1.1	192.168.35.211	DNS	116	Standard query response 0x3132 A sb-ssl.google.com CNAME sb-ssl.l.google.com
30343	56.632426	192.168.1.1	192.168.35.211	DNS	100	Standard query response 0x5159 HTTPS sb-ssl.google.com CNAME sb-ssl.l.google
33670	66.456517	192.168.35.211	192.168.1.1	DNS	74	Standard query 0x3232 A www.google.com
33671	66.456809	192.168.35.211	192.168.1.1	DNS	74	Standard query 0xa811 HTTPS www.google.com
33672	66.457079	192.168.1.1	192.168.35.211	DNS	90	Standard query response 0x3232 A www.google.com A 142.250.192.68
33673	66.457326	192.168.1.1	192.168.35.211	DNS	99	Standard query response 0xa811 HTTPS www.google.com HTTPS
34170	135.367896	192.168.35.211	192.168.1.1	DNS	76	Standard query 0x73d4 A dit.whatsapp.net
34171	135.368022	192.168.35.211	192.168.1.1	DNS	76	Standard query 0x7fd1 HTTPS dit.whatsapp.net
34172	135.372845	192.168.1.1	192.168.35.211	DNS	124	Standard query response 0x7fd1 HTTPS dit.whatsapp.net CNAME scontent.whatsap
34173	135.373909	192.168.1.1	192.168.35.211	DNS	140	Standard query response 0x73d4 A dit.whatsapp.net CNAME scontent.whatsapp.ne
34785	217.698248	192.168.35.211	192.168.1.1	DNS	89	Standard query 0xde1d A media-bom1-1.cdn.whatsapp.net
34786	217.698378	192.168.35.211	192.168.1.1	DNS	89	Standard query 0xe9dc HTTPS media-bom1-1.cdn.whatsapp.net
34787	217.699064	192.168.1.1	192.168.35.211	DNS	105	Standard query response 0xde1d A media-bom1-1.cdn.whatsapp.net A 157.240.16.
34788	217.699064	192.168.1.1	192.168.35.211	DNS	89	Standard query response 0xe9dc HTTPS media-bom1-1.cdn.whatsapp.net
38491	287.888574	192.168.35.211	192.168.1.1	DNS	73	Standard query 0xf2d5 A dl.google.com
38492	287.889097	192.168.1.1	192.168.35.211	DNS	89	Standard query response 0xf2d5 A dl.google.com A 142.250.183.46

```
▷ Frame 34170: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_{FFA154E6-A2AE-4CF3-8C23-7E71D70351C3}, id 0
▷ Ethernet II, Src: Pegatron_91:8f:e0 (4c:72:b9:91:8f:e0), Dst: Cisco_40:ca:5a (08:1f:f3:40:ca:5a)
▷ Internet Protocol Version 4, Src: 192.168.35.211, Dst: 192.168.1.1
▷ User Datagram Protocol, Src Port: 62794, Dst Port: 53
◢ Domain Name System (query)
    Transaction ID: 0x73d4
  ▷ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
    [Response In: 34173]
```

```
0000  08 1f
0010  00 3e
0020  01 01
0030  00 00
0040  61 70
```

(Reply) :

dns

No.	Time	Source	Destination	Protocol	Length	Info
30342	56.632426	192.168.1.1	192.168.35.211	DNS	116	Standard query response 0x3132 A sb-ssl.google.
30343	56.632426	192.168.1.1	192.168.35.211	DNS	100	Standard query response 0x5159 HTTPS sb-ssl.goo
33670	66.456517	192.168.35.211	192.168.1.1	DNS	74	Standard query 0x3232 A www.google.com
33671	66.456809	192.168.35.211	192.168.1.1	DNS	74	Standard query 0xa811 HTTPS www.google.com
33672	66.457079	192.168.1.1	192.168.35.211	DNS	90	Standard query response 0x3232 A www.google.com
33673	66.457326	192.168.1.1	192.168.35.211	DNS	99	Standard query response 0xa811 HTTPS www.google
34170	135.367896	192.168.35.211	192.168.1.1	DNS	76	Standard query 0x73d4 A dit.whatsapp.net
34171	135.368022	192.168.35.211	192.168.1.1	DNS	76	Standard query 0x7fd1 HTTPS dit.whatsapp.net
34172	135.372845	192.168.1.1	192.168.35.211	DNS	124	Standard query response 0x7fd1 HTTPS dit.whatsa
34173	135.373909	192.168.1.1	192.168.35.211	DNS	140	Standard query response 0x73d4 A dit.whatsapp.n
34785	217.698248	192.168.35.211	192.168.1.1	DNS	89	Standard query 0xde1d A media-bom1-1.cdn.whatsa
34786	217.698378	192.168.35.211	192.168.1.1	DNS	89	Standard query 0xe9dc HTTPS media-bom1-1.cdn.wh
34787	217.699064	192.168.1.1	192.168.35.211	DNS	105	Standard query response 0xde1d A media-bom1-1.c
34788	217.699064	192.168.1.1	192.168.35.211	DNS	89	Standard query response 0xe9dc HTTPS media-bom1
38491	287.888574	192.168.35.211	192.168.1.1	DNS	73	Standard query 0xf2d5 A dl.google.com
38492	287.889097	192.168.1.1	192.168.35.211	DNS	89	Standard query response 0xf2d5 A dl.google.com

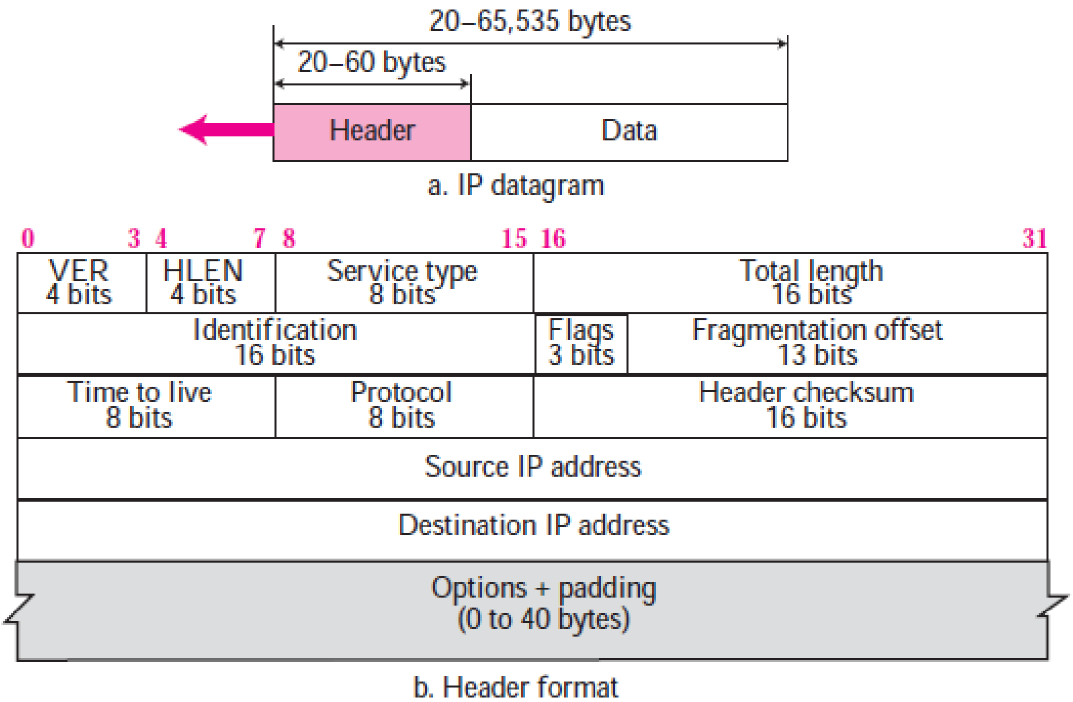
```
▷ Frame 34172: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface \Device\NPF_{FFA154E6-A2AE-4CF3-8C23-
▷ Ethernet II, Src: Cisco_40:ca:5a (08:1f:f3:40:ca:5a), Dst: Pegatron_91:8f:e0 (4c:72:b9:91:8f:e0)
▷ Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.35.211
▷ User Datagram Protocol, Src Port: 53, Dst Port: 57901
◢ Domain Name System (response)
    Transaction ID: 0x7fd1
  ▷ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
  ▷ Answers
    [Request In: 34171]
    [Time: 0.004823000 seconds]
```

Experiment No. 6

Objective: To investigate Internet Protocol for IPV4 using Wireshark.

Introduction

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



Here is a description of each field:

- **Version** – the version of the IP protocol. For IPv4, this field has a value of 4.
- **Header length** – the length of the header in 32-bit words. The minumum value is 20 bytes, and the maximum value is 60 bytes.
- **Priority and Type of Service** – specifies how the datagram should be handled. The first 3 bits are the priority bits.
- **Total length** – the length of the entire packet (header + data). The minimum length is 20 bytes, and the maximum is 65,535 bytes.
- **Identification** – used to differentiate fragmented packets from different datagrams.
- **Flags** – used to control or identify fragments.
- **Fragmented offset** – used for fragmentation and reassembly if the packet is too large to put in a frame.

It is obvious that even if each fragment follows a different path and arrives out of order, the final destination host can reassemble the original datagram from the fragments received (if none of them is lost) using the following strategy:

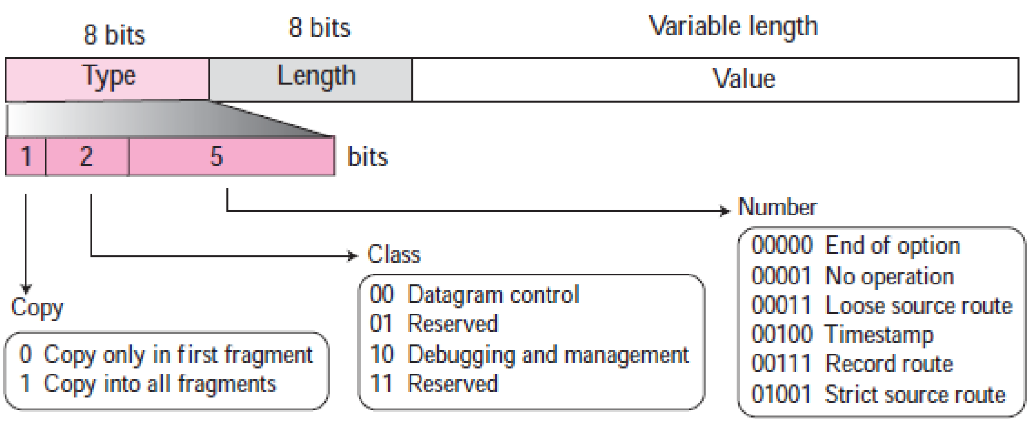
a. The first fragment has an offset field value of zero.

b. Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.

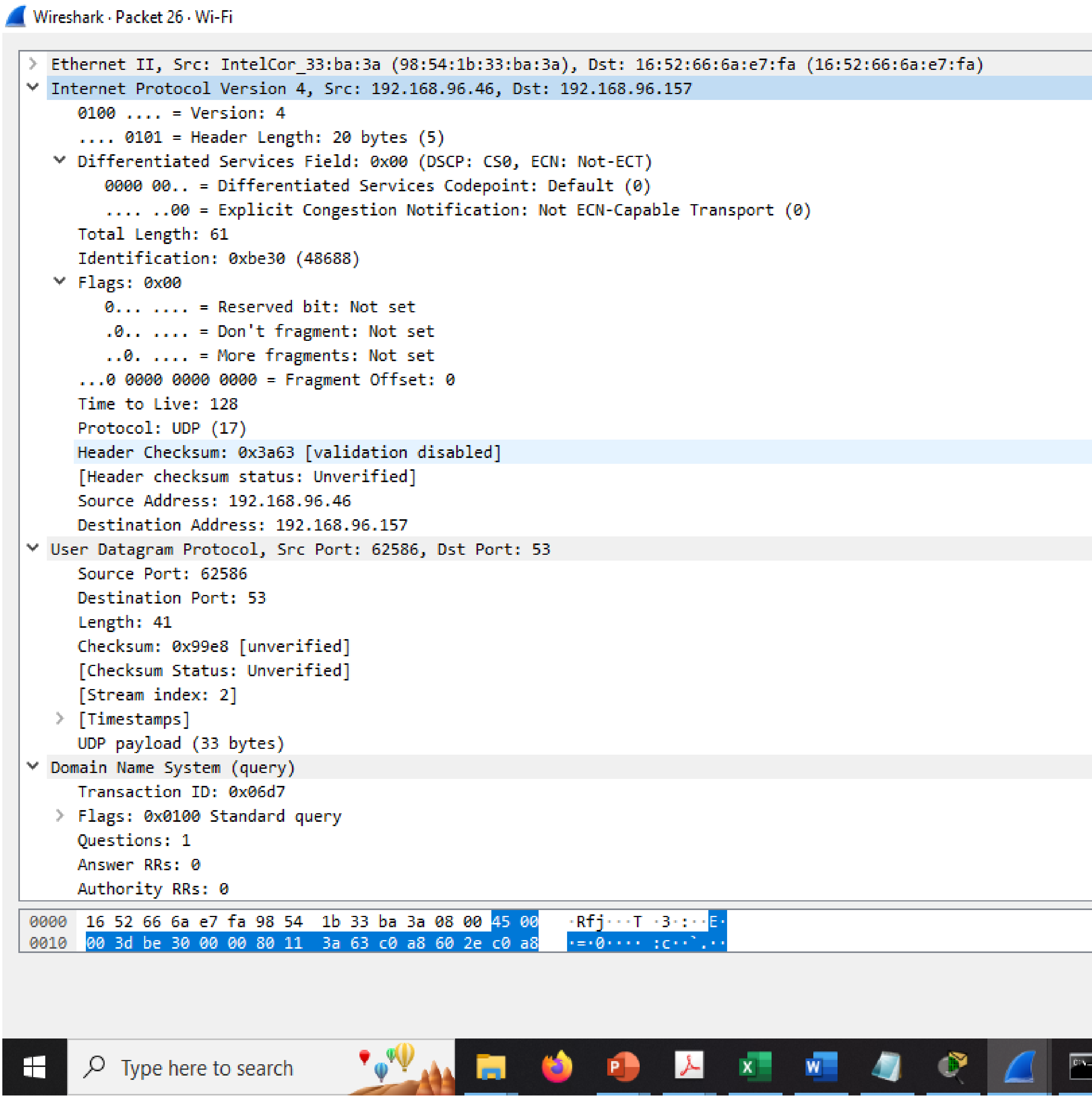
c. Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.

d. Continue the process. The last fragment has a more bit value of 0.

- **Time to live** – limits a datagram's lifetime. If the packet doesn't get to its destination before the TTL expires, it is discarded.
- **Protocol** – defines the protocol used in the data portion of the IP datagram. For example, TCP is represented by the number 6 and UDP by 17.
- **Header checksum** – used for error-checking of the header. If a packet arrives at a router and the router calculates a different checksum than the one specified in this field, the packet will be discarded.
- **Source IP address** – the IP address of the host that sent the packet.
- **Destination IP address** – the IP address of the host that should receive the packet.
- **Options** – used for network testing, debugging, security, and more.



Notice the fields in the header: the IP version is IPv4, the header length is 20 bytes, the upper-level protocol used is TCP, the TTL value is set tu 128, source and destination IP addresses are listed, etc.



Q: Add screenshot of any IP Packet you grabbed in wireshark and identify the IP addresses used

Experiment No. 7

Objective: To investigate TCP protocol using Wireshark.

Introduction:

TCP stands for **Transmission Control Protocol**. It is a transport layer protocol that facilitates the transmission of packets from source to destination. It is a connection-oriented protocol that means it establishes the connection prior to the communication that occurs between the computing devices in a network. This protocol is used with an IP protocol, so together, they are referred to as a TCP/IP.

The main functionality of the TCP is to take the data from the application layer. Then it divides the data into a several packets, provides numbering to these packets, and finally transmits these packets to the destination. The TCP, on the other side, will reassemble the packets and transmits them to the application layer. As we know that TCP is a connection-oriented protocol, so the connection will remain established until the communication is not completed between the sender and the receiver.

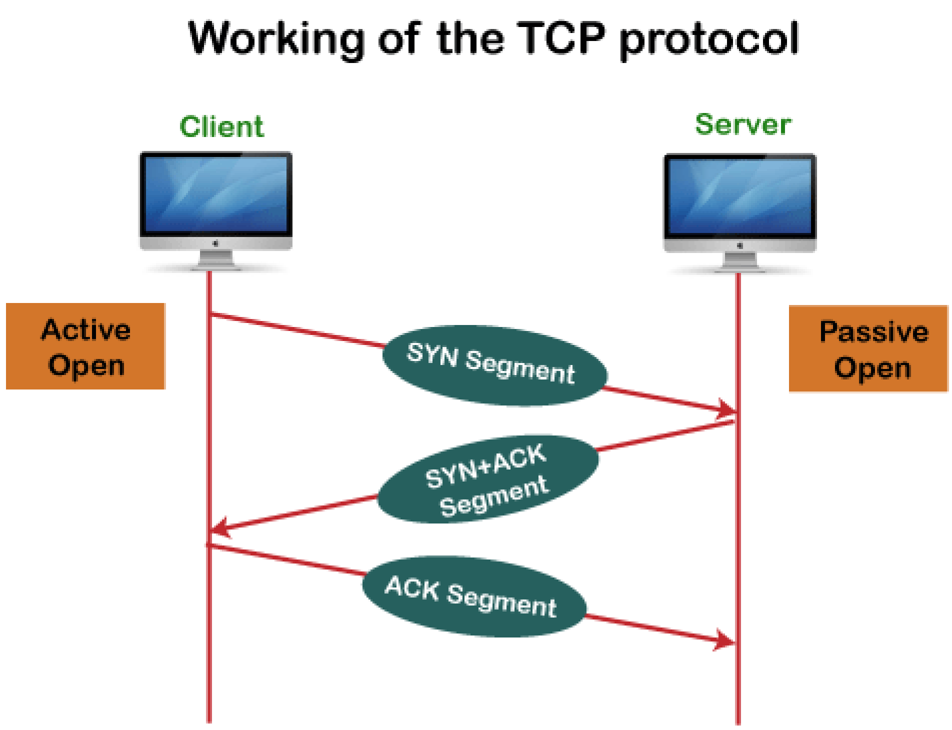
Features of TCP protocol

The following are the features of a TCP protocol:

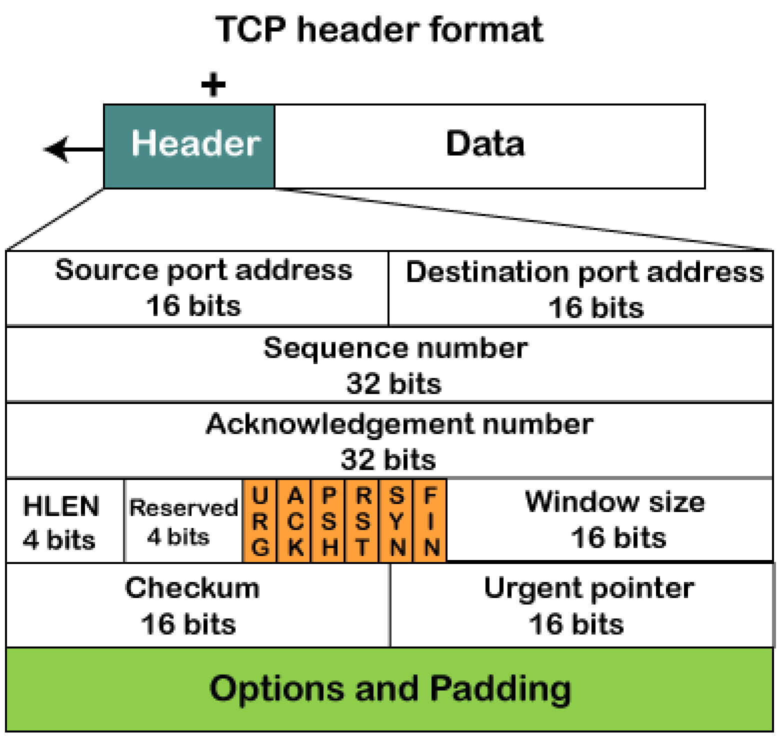
- **Transport Layer Protocol**
- **Reliable**
- **Order of the data is maintained**
- **Connection-oriented**
- **Full duplex**
- **Stream-oriented**

Working of TCP

In TCP, the connection is established by using three-way handshaking. The client sends the segment with its sequence number. The server, in return, sends its segment with its own sequence number as well as the acknowledgement sequence, which is one more than the client sequence number. When the client receives the acknowledgment of its segment, then it sends the acknowledgment to the server. In this way, the connection is established between the client and the server.



TCP Header format



- **Source port:** It defines the port of the application, which is sending the data. So, this field contains the source port address, which is 16 bits.
- **Destination port:** It defines the port of the application on the receiving side. So, this field contains the destination port address, which is 16 bits.
- **Sequence number:** This field contains the sequence number of data bytes in a particular session.
- **Acknowledgment number:** When the ACK flag is set, then this contains the next sequence number of the data byte and works as an acknowledgment for the previous data received. For example, if the receiver receives the segment number 'x', then it responds 'x+1' as an acknowledgment number.
- **HLEN:** It specifies the length of the header indicated by the 4-byte words in the header. The size of the header lies between 20 and 60 bytes. Therefore, the value of this field would lie between 5 and 15.
- **Reserved:** It is a 4-bit field reserved for future use, and by default, all are set to zero.
- **Flags**

  There are six control bits or flags:

  1. **URG:** It represents an urgent pointer. If it is set, then the data is processed urgently.
  2. **ACK:** If the ACK is set to 0, then it means that the data packet does not contain an acknowledgment.
  3. **PSH:** If this field is set, then it requests the receiving device to push the data to the receiving application without buffering it.
  4. **RST:** If it is set, then it requests to restart a connection.
  5. **SYN:** It is used to establish a connection between the hosts.
  6. **FIN:** It is used to release a connection, and no further data exchange will happen.

- **Window size**

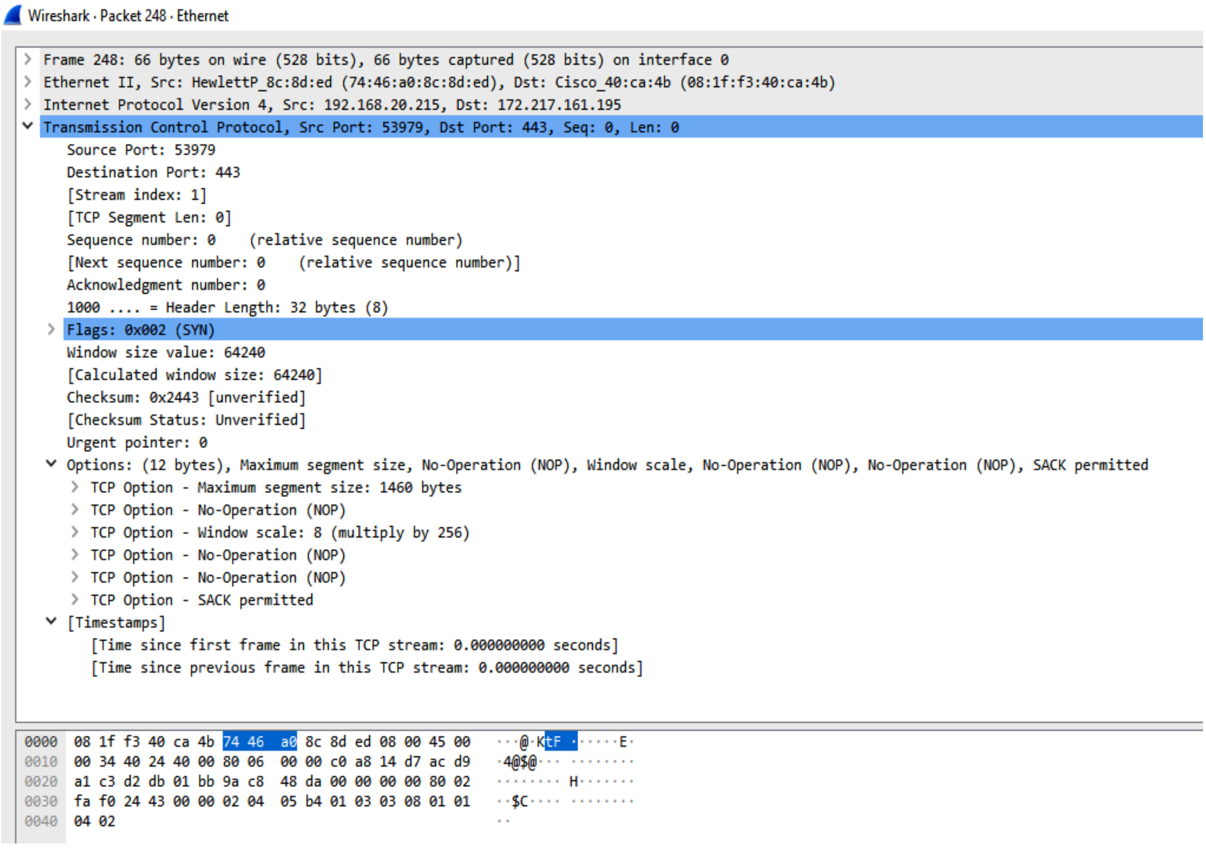
  It is a 16-bit field. It contains the size of data that the receiver can accept. This field is used for the flow control between the sender and receiver and also determines the amount of buffer allocated by the receiver for a segment. The value of this field is determined by the receiver.
- **Checksum**

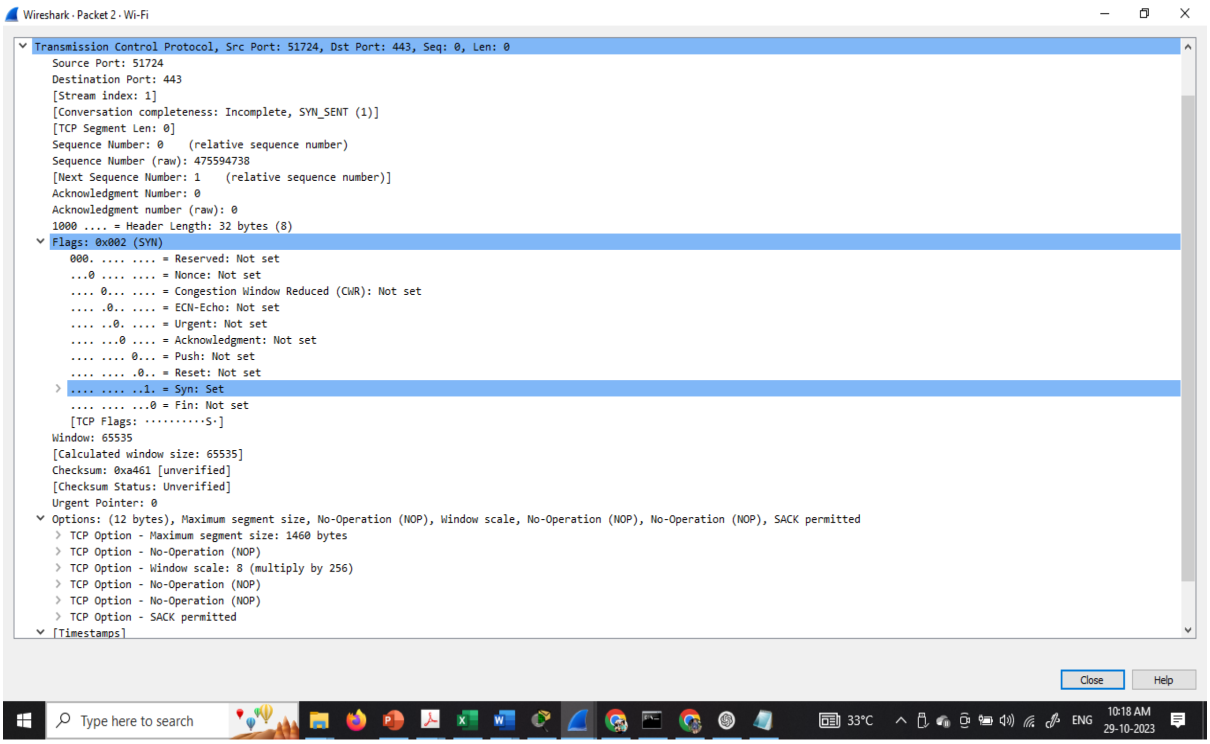
  It is a 16-bit field. This field is optional in UDP, but in the case of TCP/IP, this field is mandatory.
- **Urgent pointer**

  It is a pointer that points to the urgent data byte if the URG flag is set to 1. It defines a value that will be added to the sequence number to get the sequence number of the last urgent byte.
- **Options**

  It provides additional options. The optional field is represented in 32-bits. If this field contains the data less than 32-bit, then padding is required to obtain the remaining bits.

Output


Wireshark · Packet 248 · Ethernet
Frame 248: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: HewlettP_8c:8d:ed (74:46:a0:8c:8d:ed), Dst: Cisco_40:ca:4b (08:1f:f3:40:ca:4b)
Internet Protocol Version 4, Src: 192.168.20.215, Dst: 172.217.161.195
Transmission Control Protocol, Src Port: 53979, Dst Port: 443, Seq: 0, Len: 0
Source Port: 53979
Destination Port: 443
[Stream index: 1]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
[Next sequence number: 0 (relative sequence number)]
Acknowledgment number: 0
1000 .... = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
Window size value: 64240
[Calculated window size: 64240]
Checksum: 0x2443 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
TCP Option - Maximum segment size: 1460 bytes
TCP Option - No-Operation (NOP)
TCP Option - Window scale: 8 (multiply by 256)
TCP Option - No-Operation (NOP)
TCP Option - No-Operation (NOP)
TCP Option - SACK permitted
[Timestamps]
[Time since first frame in this TCP stream: 0.000000000 seconds]
[Time since previous frame in this TCP stream: 0.000000000 seconds]
0000 08 1f f3 40 ca 4b 74 46 a0 8c 8d ed 08 00 45 00
0010 00 34 40 24 40 00 80 06 00 00 c0 a8 14 d7 ac d9
0020 a1 c3 d2 db 01 bb 9a c8 48 da 00 00 00 00 80 02
0030 fa f0 24 43 00 00 02 04 05 b4 01 03 03 08 01 01
0040 04 02


Wireshark · Packet 2 · Wi-Fi
Transmission Control Protocol, Src Port: 51724, Dst Port: 443, Seq: 0, Len: 0
Source Port: 51724
Destination Port: 443
[Stream index: 1]
[Conversation completeness: Incomplete, SYN_SENT (1)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 475594738
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1000 .... = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
000. .... = Reserved: Not set
...0 .... = Nonce: Not set
.... 0... .... = Congestion Window Reduced (CWR): Not set
.... .0.. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
....0 .... = Acknowledgment: Not set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
....1. = Syn: Set
....0 = Fin: Not set
[TCP Flags: ··········S·]
Window: 65535
[Calculated window size: 65535]
Checksum: 0xa461 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
TCP Option - Maximum segment size: 1460 bytes
TCP Option - No-Operation (NOP)
TCP Option - Window scale: 8 (multiply by 256)
TCP Option - No-Operation (NOP)
TCP Option - No-Operation (NOP)
TCP Option - SACK permitted
[Timestamps]
Close
Help
Type here to search
33°C
ENG
10:18 AM
29-10-2023

Experiment No. 8

Objective: To investigate ICMP protocol using Wireshark.

Introduction :

The ICMP stands for Internet Control Message Protocol. It is a network layer protocol. It is used for error handling in the network layer, and it is primarily used on network devices such as routers. As different types of errors can exist in the network layer, so ICMP can be used to report these errors and to debug those errors.

For example, some sender wants to send the message to some destination, but the router couldn't send the message to the destination. In this case, the router sends the message to the sender that I could not send the message to that destination.

The IP protocol does not have any error-reporting or error-correcting mechanism, so it uses a message to convey the information. For example, if someone sends the message to the destination, the message is somehow stolen between the sender and the destination. If no one reports the error, then the sender might think that the message has reached the destination. If someone in-between reports the error, then the sender will resend the message very quickly.

Messages

The ICMP messages are usually divided into two categories:

ICMP messages

Category	Type	Message
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

- **Error-reporting messages**

The error-reporting message means that the router encounters a problem when it processes an IP packet then it reports a message.

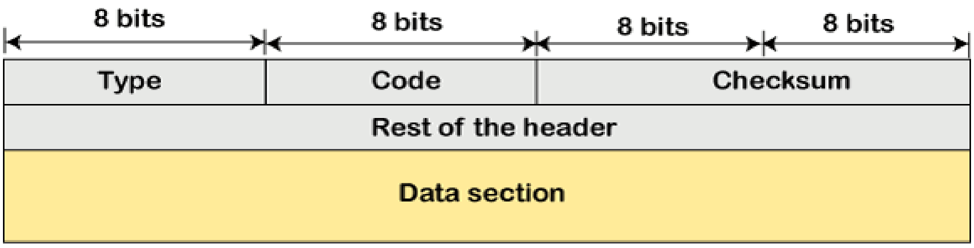
- **Query messages**

The query messages are those messages that help the host to get the specific information of another host. For example, suppose there are a client and a server, and the client wants to know whether the server is live or not, then it sends the ICMP message to the server.

ICMP Message Format

The message format has two things; one is a category that tells us which type of message it is. If the message is of error type, the error message contains the type and the code. The type defines the type of message while the code defines the subtype of the message.

The ICMP message contains the following fields:



- **Type:** It is an 8-bit field. It defines the ICMP message type. The values range from 0 to 127 are defined for ICMPv6, and the values from 128 to 255 are the informational messages.
- **Code:** It is an 8-bit field that defines the subtype of the ICMP message
- **Checksum:** It is a 16-bit field to detect whether the error exists in the message or not.

❑ The Internet Control Message Protocol (ICMP) supports the unreliable and connectionless Internet Protocol (IP).

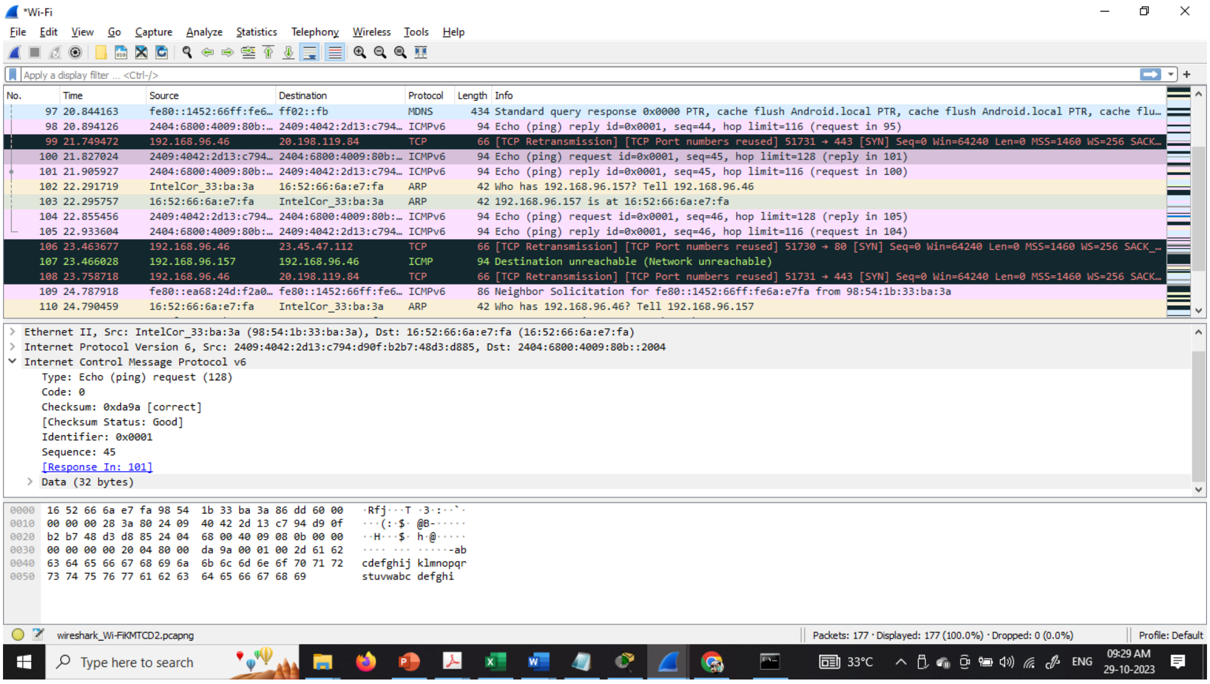
❑ ICMP messages are encapsulated in IP datagrams. There are two categories of ICMP messages: error-reporting and query messages. The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet. The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.

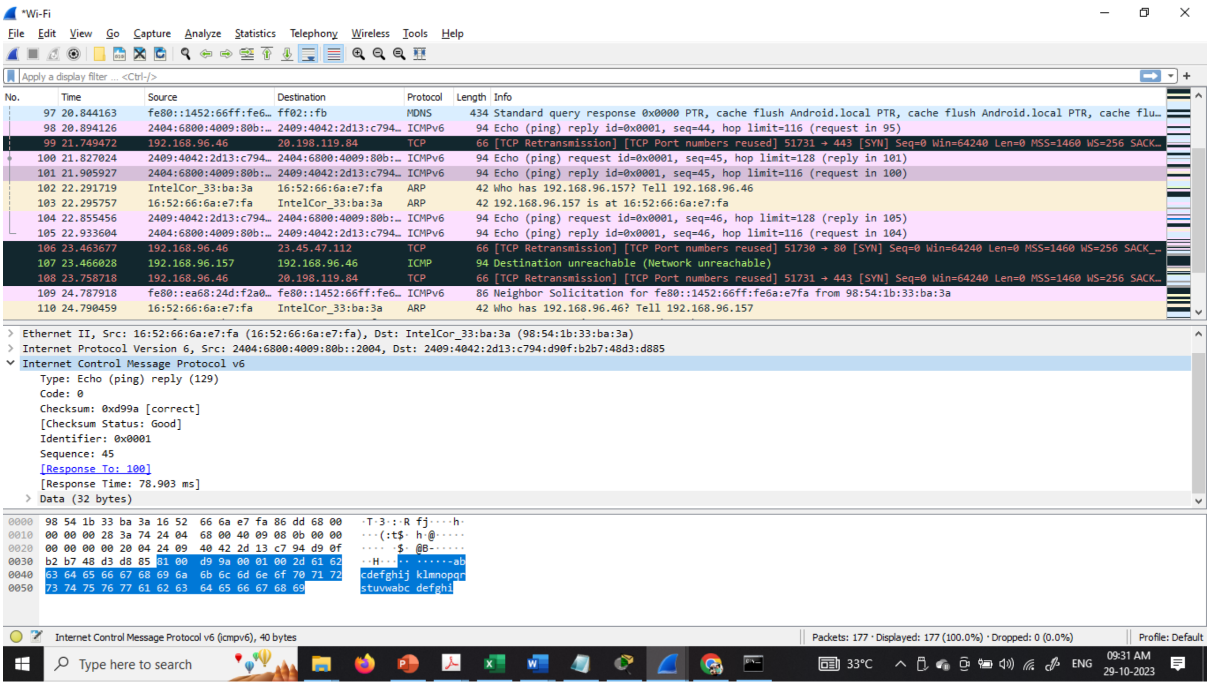
❑ The checksum for ICMP is calculated using both the header and the data fields of the ICMP message.

❑ There are several tools that can be used in the Internet for debugging. We can

find if a host or router is alive and running. Two of these tools are ping and traceroute.

❑ A simple ICMP design can consist of an input module that handles incoming ICMP packets and an output module that handles demands for ICMP services.





Q: Add screenshot of any ICMP Packet you grabbed in wireshark when tracert command is getting executed

Experiment No. 9

Objective: The objective of this experiment is to familiarize students with common networking commands used in Windows Command Prompt for troubleshooting and managing network connections.

ipconfig Command:

Use the ipconfig command to display the IP configuration of your computer. Note down the IP address, subnet mask, default gateway, and DNS server addresses.

Example command syntax

ipconfig

ping Command:

Use the ping command to test the reachability of a website. For example, ping Google's DNS server.

Example command syntax

ping 8.8.8.8

Observe the response times and packet loss (if any).

tracert Command:

Use the tracert command to trace the route to a website, e.g., google.com.

Example command syntax

tracert google.com

Take note of the IP addresses of each hop along the route.

nslookup Command:

Use the nslookup command to perform a DNS query for a domain name, e.g., yahoo.com.

Example command syntax

nslookup yahoo.com

Record the IP addresses and the DNS server used for the query.

arp Command:

Use the arp command to display the ARP cache.

Example command syntax

arp -a

Note down the IP-to-MAC address mappings.

netstat Command:

Use the netstat command to display active network connections and listening ports. Example command syntax

netstat -ano

Identify active connections and associated process IDs.

route Command:

Use the route print command to display the local IP routing table.

arduino

Example command syntax

route print

Observe the routing information.

ipconfig /flushdns Command:

Use the ipconfig /flushdns command to flush the DNS resolver cache.

Example command syntax

ipconfig /flushdns

Confirm the cache has been flushed.

ipconfig /renew Command:

Use the ipconfig /renew command to renew the DHCP configuration.

Example command syntax

ipconfig /renew

Observe the changes in the IP configuration.

Conclusion:

In this experiment, you learned how to use various networking commands in the Windows Command Prompt to gather information about network connections, resolve DNS queries, and troubleshoot network-related issues.

Q: Add screenshots of networking commands executed on your machine

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping www.google.com

Pinging www.google.com [2404:6800:4009:828::2004] with 32 bytes of data:
Reply from 2404:6800:4009:828::2004: time=58ms
Reply from 2404:6800:4009:828::2004: time=66ms
Reply from 2404:6800:4009:828::2004: time=66ms
Reply from 2404:6800:4009:828::2004: time=53ms

Ping statistics for 2404:6800:4009:828::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 53ms, Maximum = 66ms, Average = 60ms

C:\Windows\system32>ping 192.168.96.157

Pinging 192.168.96.157 with 32 bytes of data:
Reply from 192.168.96.157: bytes=32 time=1ms TTL=64
Reply from 192.168.96.157: bytes=32 time=1ms TTL=64
Reply from 192.168.96.157: bytes=32 time=24ms TTL=64
Reply from 192.168.96.157: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.96.157:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 24ms, Average = 7ms

C:\Windows\system32>ping 192.168.35.12

Pinging 192.168.35.12 with 32 bytes of data:
Reply from 192.168.35.12: bytes=32 time<1ms TTL=128
Reply from 192.168.35.12: bytes=32 time<1ms TTL=128
Reply from 192.168.35.12: bytes=32 time<1ms TTL=128
Reply from 192.168.35.12: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.35.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Windows\system32>
```

Administrator: Command Prompt

```
C:\Windows\system32>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : VipulSir
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No


Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 3165
   Physical Address. . . . . . . . . : 98-54-1B-33-BA-3A
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2409:4042:2d13:c794:d15a:b697:ea7f:d380(Preferred)
   Temporary IPv6 Address. . . . . . : 2409:4042:2d13:c794:28ea:4b0d:2edc:b6b5(Preferred)
   Link-local IPv6 Address . . . . . : fe80::ea68:24d:f2a0:6fde%18(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.96.46(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 29 October 2023 08:31:51
   Lease Expires . . . . . . . . . . : 29 October 2023 10:38:00
   Default Gateway . . . . . . . . . : fe80::1452:66ff:fe6a:e7fa%18
                                       192.168.96.157
   DHCP Server . . . . . . . . . . . : 192.168.96.157
   DHCPv6 IAID . . . . . . . . . . . : 530076699
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-0C-F9-40-A8-1E-84-15-EA-04
   DNS Servers . . . . . . . . . . . : 192.168.96.157
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Administrator: Command Prompt - netstat

```
C:\Windows\system32>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49671        VipulSir:49672         ESTABLISHED
  TCP    127.0.0.1:49672        VipulSir:49671         ESTABLISHED
  TCP    127.0.0.1:49675        VipulSir:49676         ESTABLISHED
  TCP    127.0.0.1:49676        VipulSir:49675         ESTABLISHED
  TCP    127.0.0.1:49677        VipulSir:49680         ESTABLISHED
  TCP    127.0.0.1:49678        VipulSir:49679         ESTABLISHED
  TCP    127.0.0.1:49679        VipulSir:49678         ESTABLISHED
  TCP    127.0.0.1:49680        VipulSir:49677         ESTABLISHED
  TCP    127.0.0.1:49681        VipulSir:49686         ESTABLISHED
  TCP    127.0.0.1:49686        VipulSir:49681         ESTABLISHED
  TCP    127.0.0.1:52649        VipulSir:52650         ESTABLISHED
  TCP    127.0.0.1:52650        VipulSir:52649         ESTABLISHED
  TCP    192.168.96.46:53110    152.195.38.76:http     SYN_SENT
  TCP    192.168.96.46:53111    152.195.38.76:http     SYN_SENT
  TCP    192.168.96.46:53117    152.195.38.76:http     SYN_SENT
  TCP    192.168.96.46:53119    104.208.16.88:https    SYN_SENT
```

Administrator: Command Prompt

```
C:\Windows\system32>tracert www.facebook.com

Tracing route to star-mini.c10r.facebook.com [2a03:2880:f12f:183:face:b00c:0:25de]
over a maximum of 30 hops:

  1     2 ms     1 ms     1 ms  2409:4042:2d13:c794::12
  2     *        *        *     Request timed out.
  3    78 ms    48 ms    38 ms  2405:200:381:eeee:20::1778
  4   121 ms    39 ms    48 ms  2405:200:801:c00::17aa
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7    73 ms    91 ms    54 ms  ae5.pr02.bom1.tfbnw.net [2620:0:1cff:dead:beee::6a]
  8   125 ms   117 ms    64 ms  ae5.pr02.bom1.tfbnw.net [2620:0:1cff:dead:beee::6a]
  9    66 ms    48 ms   104 ms  po102.psw04.bom1.tfbnw.net [2620:0:1cff:dead:bef0::185]
 10   140 ms    62 ms    56 ms  po8.msw1aq.02.bom1.tfbnw.net [2a03:2880:f02f:ffff::38d]
 11   143 ms    54 ms    46 ms  edge-star-mini6-shv-02-bom1.facebook.com [2a03:2880:f12f:183:face:b00c:0:25de]

Trace complete.

C:\Windows\system32>tracert www.google.com

Tracing route to www.google.com [2404:6800:4009:826::2004]
over a maximum of 30 hops:

  1     3 ms     2 ms     2 ms  2409:4042:2d13:c794::12
  2     *        *        *     Request timed out.
  3    85 ms    95 ms    41 ms  2405:200:381:eeee:20::1778
  4   148 ms    52 ms   110 ms  2405:200:801:c00::17a8
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7    73 ms    64 ms    57 ms  2001:4860:1:1::3c8
  8    72 ms    58 ms    56 ms  2404:6800:80b2::1
  9   211 ms    84 ms    63 ms  2001:4860:0:1::27e6
 10    96 ms    91 ms    55 ms  2001:4860:0:115c::3
 11     *        *      101 ms  2001:4860::12:0:b974
 12    55 ms    82 ms    58 ms  2001:4860:0:1::5429
 13    57 ms    69 ms    55 ms  bom07s33-in-x04.1e100.net [2404:6800:4009:826::2004]

Trace complete.
```