

Module 7 and 8

Q1. Try Test-Connection and nslookup commands for below websites

www.google.com

www.facebook.com

www.amazon.com

www.github.com

www.cisco.com

Test-Connection: This command checks if a website is reachable.

nslookup: This retrieves the IP address of the domain,i.e., finds the ip address of the website.

```
PS C:\Users\admin> Test-Connection www.google.com -Count 4
```

Source	Destination	IPv4Address	IPv6Address
DINESH	www.google.com	142.250.194.196	2404:6800:4002:824::2004
DINESH	www.google.com	142.250.194.196	2404:6800:4002:824::2004
DINESH	www.google.com	142.250.194.196	2404:6800:4002:824::2004
DINESH	www.google.com	142.250.194.196	2404:6800:4002:824::2004

```
PS C:\Users\admin> Test-Connection www.facebook.com -Count 4
```

Source	Destination	IPv4Address	IPv6Address
DINESH	www.facebook...	163.70.139.35	2a03:2880:f184:186:fa...
DINESH	www.facebook...	163.70.139.35	2a03:2880:f184:186:fa...
DINESH	www.facebook...	163.70.139.35	2a03:2880:f184:186:fa...
DINESH	www.facebook...	163.70.139.35	2a03:2880:f184:186:fa...

```
PS C:\Users\admin> Test-Connection www.cisco.com -Count 4
```

Source	Destination	IPv4Address	IPv6Address
DINESH	www.cisco.com	23.10.232.78	2600:1417:2c:19e::b33
DINESH	www.cisco.com	23.10.232.78	2600:1417:2c:19e::b33
DINESH	www.cisco.com	23.10.232.78	2600:1417:2c:19e::b33
DINESH	www.cisco.com	23.10.232.78	2600:1417:2c:19e::b33

```
PS C:\Users\admin> Test-Connection www.github.com -Count 4
```

Source	Destination	IPv4Address	IPv6Address
DINESH	www.github.com	20.207.73.82	
DINESH	www.github.com	20.207.73.82	
DINESH	www.github.com	20.207.73.82	
DINESH	www.github.com	20.207.73.82	

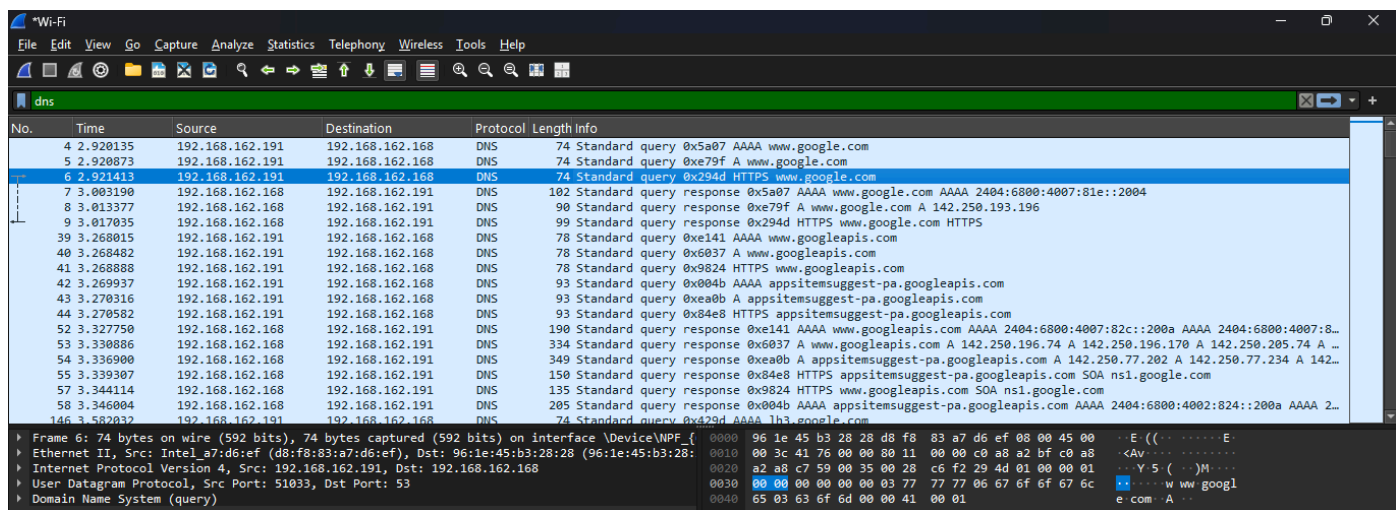
```
PS C:\Users\admin> Test-Connection www.amazon.com -Count 4
```

Source	Destination	IPv4Address	IPv6Address
DINESH	www.amazon.com	52.84.241.156	2600:9000:2656:5a00:7...
DINESH	www.amazon.com	52.84.241.156	2600:9000:2656:5a00:7...
DINESH	www.amazon.com	52.84.241.156	2600:9000:2656:5a00:7...
DINESH	www.amazon.com	52.84.241.156	2600:9000:2656:5a00:7...

Q2. Use Wireshark to capture and analyze DNS, TCP, UDP traffic and packet header, packet flow, options and flags

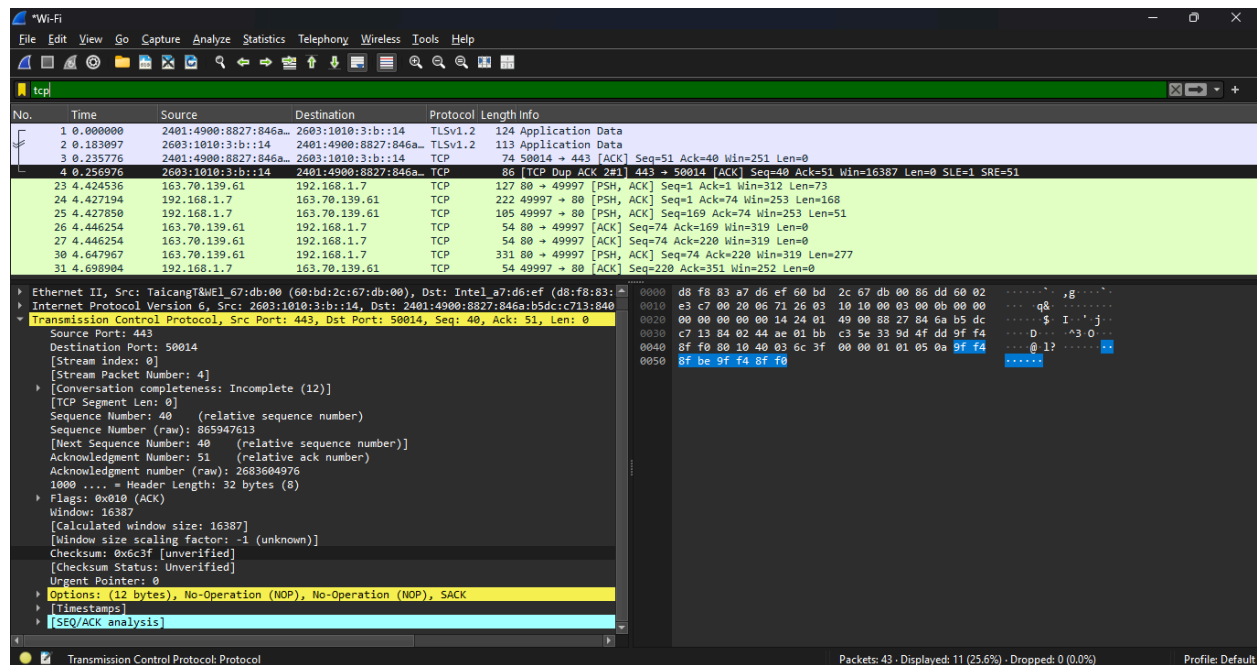
DNS protocol:

- The Domain Name System (DNS) protocol is a network protocol that translates domain names into IP addresses.
- To capture DNS packets, I searched www.google.com
- As a Response it Shows the IP address for www.google.com which is 142.250.193.196.



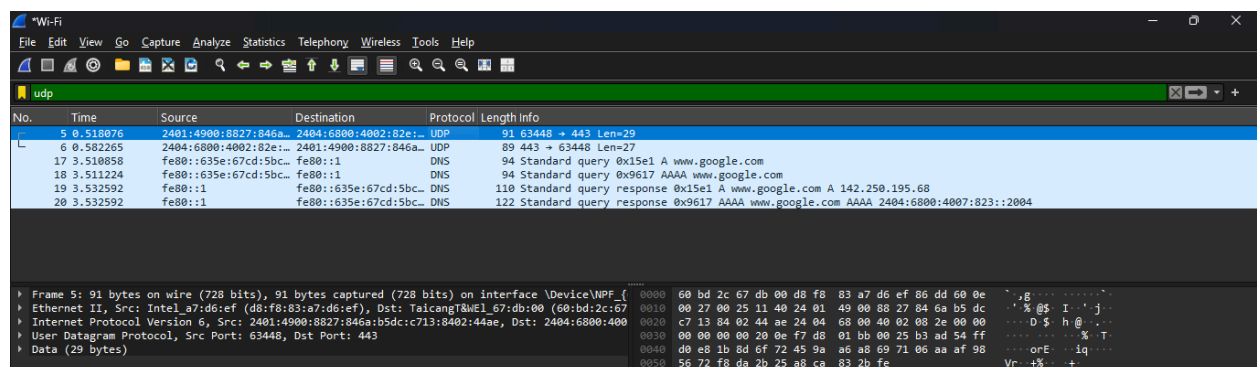
TCP:

- Transmission Control Protocol is a connection-oriented protocol for communications that helps in the exchange of messages between different devices over a network.
- TCP establishes a reliable connection between sender and receiver using the three-way handshake (SYN, SYN-ACK, ACK) and it uses a four-step handshake (FIN, ACK, FIN, ACK) to close connections properly.



UDP:

- User Datagram Protocol is a connectionless and light-weight protocol and thus it is unreliable.



Q3. Explore traceroute/tracert for different websites eg:google.com and analyse the parameters in the output and explore different options for traceroute command

Traceroute :

- **tracert** is the windows version which is a network diagnostic tool used to trace the path packets take from your device to a destination (e.g., google.com). They help identify routing issues and checks whether the packet has reached the destination address.
- It provides a detailed path taken by the packets from the source device to the destination, showing each intermediate router or "hop" along the way.

There are various versions of tracert which are as follows:

- **tracert www.google.com** : Basic syntax to run tracert to a specified destination (IP address or domain name).

```
C:\Users\admin>tracert www.google.com

Tracing route to www.google.com [2404:6800:4009:82e::2004]
over a maximum of 30 hops:

  1    3 ms    2 ms    2 ms    2401:4900:8827:846a::1
  2    *        *        *        Request timed out.
  3   34 ms    *       130 ms   fc00::92
  4   15 ms   15 ms   15 ms   2404:a800:3a00:300::85
  5   32 ms   26 ms   24 ms   2404:a800::92
  6   36 ms   34 ms   30 ms   2001:4860:1:1::674
  7   31 ms    *        *       2404:6800:8202:1c0::1
  8   31 ms   29 ms   27 ms   2001:4860:0:1::56e
  9   26 ms   25 ms   28 ms   2001:4860:0:1::4878
 10   48 ms   48 ms   44 ms   2001:4860::9:4001:7734
 11   44 ms   42 ms   44 ms   2001:4860:0:1::87b3
 12   43 ms   44 ms   47 ms   2001:4860:0:1::2131
 13   45 ms   44 ms   42 ms   bom07s37-in-x04.1e100.net [2404:6800:4009:82e::2004]

Trace complete.
```

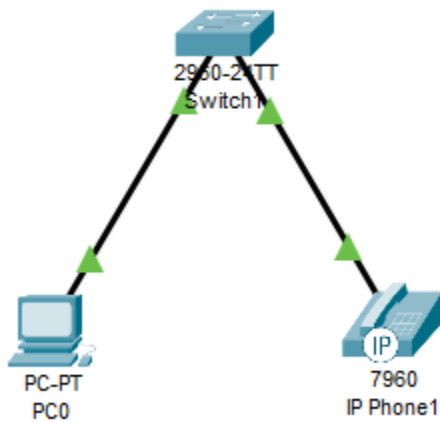
- **tracert -h 20 www.google.com** : Sets the maximum number of hops to trace before terminating.
- **tracert -w 1000 www.google.com** : Specifies the timeout in milliseconds to wait for each reply. Default is 4000 ms.
- **tracert -4 www.google.com** : Forces tracert to use IPv4 for the trace.
- **tracert -6 www.google.com** : Forces tracert to use IPv6 for the trace.
- **tracert -R www.google.com** : This option routes each hop in reverse, from destination back to the source.

Use Cisco packet tracer for the below

Q4. Set up trunk ports between switches and try ping between different VLANs.

```
Switch(config-if)#interface GigabitEthernet0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20
Switch(config-if)#ex
Switch(config)#
```

Q7. You have a Cisco switch and a VoIP phone that needs to be placed in a voice VLAN (VLAN 20). The data for the PC should remain in a separate VLAN (VLAN 10). Configure the switch port to support both voice and data traffic.



Two VLANs are made: VLAN 10 and VLAN 20.

Configuring switch:

```
Switch>enable
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name DATA
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name VOICE
Switch(config-vlan)#exit
Switch(config)#interface fastEthernet0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#interface fastEthernet0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#show vlan
```

VLAN	Name	Status	Ports
1	default	active	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	DATA	active	Fa0/1
20	VOICE	active	Fa0/2
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Q8. You configured VLANs 10 and 20 on your switch and assigned ports to each VLAN. However, devices in VLAN 10 cannot communicate with devices in VLAN 20. Troubleshoot the issue.

If devices in VLAN 10 cannot communicate with devices in VLAN 20, the problem is likely due to the absence of inter-VLAN routing, incorrect VLAN configurations, or trunking misconfigurations. Follow these steps to diagnose and resolve the issue:

Enable Inter-VLAN Routing

Layer 2 switches cannot perform inter-VLAN routing, so a Layer 3 device (Router-on-a-Stick or Layer 3 switch) is needed.

For Router-on-a-Stick (Router with Subinterfaces):

```
Router(config)# interface GigabitEthernet 0/0.10
Router(config-subif)# encapsulation dot1Q 10
Router(config-subif)# ip address 192.168.10.1 255.255.255.0
Router(config-subif)# exit
```

```
Router(config)# interface GigabitEthernet 0/0.20
Router(config-subif)# encapsulation dot1Q 20
Router(config-subif)# ip address 192.168.20.1 255.255.255.0
Router(config-subif)# exit
```

Ensure that trunking is enabled between the switch and router.

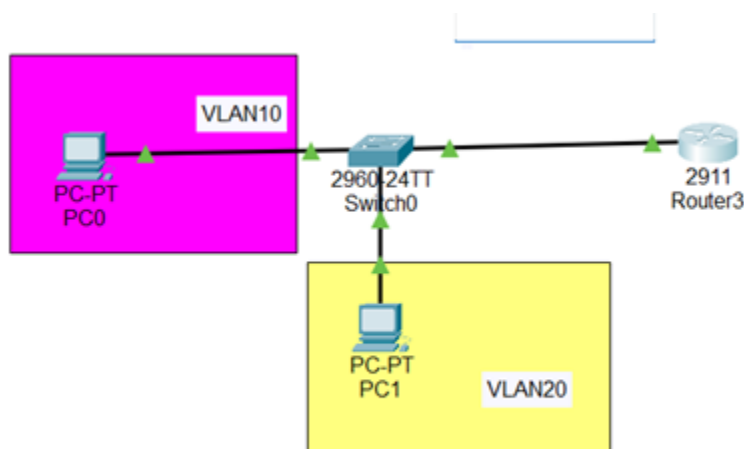
Q9. Try Inter VLAN routing with Router

- Inter-VLAN routing allows communication between different VLANs using a router.
- Since Layer 2 switches cannot route traffic between VLANs, we use a Router-on-a-Stick configuration.

Steps:

- First I have configured the switches and created VLAN10 and VLAN20.
- I connected a trunk port with the router.
- Next, I configured the router.
- I verified their connectivity using the ping command (VLAN10(PC0) to VLAN20(PC1)) and got successful replies.

Network setup:



Configuring switch:

```

Switch>en
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name VLAN10
Switch(config-vlan)#ex
Switch(config)#vlan 20
Switch(config-vlan)#name VLAN20
Switch(config-vlan)#ex
Switch(config)#interface FastEthernet 0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#ex
Switch(config)#interface FastEthernet 0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#ex
Switch(config)#

```

Trunk port connection to router:

```

Switch(config-if)#interface GigabitEthernet0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20
Switch(config-if)#ex
Switch(config)#

```

VLAN Table of switch:

```

Switch#en
Switch#show vlan brief

```

VLAN	Name	Status	Ports
1	default	active	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	VLAN10	active	Fa0/1
20	VLAN20	active	Fa0/2
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Configuring router:

```
Router#en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface GigabitEthernet 0/0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface GigabitEthernet 0/0.10
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.10, changed state to up

Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface GigabitEthernet 0/0.20
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.20, changed state to up

Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#
```

Verifying connectivity(PC0 to PC1)

```
C:\>ping 192.168.20.10

Pinging 192.168.20.10 with 32 bytes of data:

Reply from 192.168.20.10: bytes=32 time<1ms TTL=127
Reply from 192.168.20.10: bytes=32 time=1ms TTL=127
Reply from 192.168.20.10: bytes=32 time<1ms TTL=127
Reply from 192.168.20.10: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.20.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Q10. Implement ACLs to restrict traffic based on source and destination ports. Test rules by simulating legitimate and unauthorized traffic.

Access Control Lists (ACLs) are used to filter network traffic based on IP addresses, protocols, and port numbers. There are 2 ways to implement Standard ACLs (Source IP-based) or Extended ACLs (IP + Port-based).

Here, I'm using an Extended ACL to restrict traffic based on source and destination ports and test it with both legitimate and unauthorized traffic.

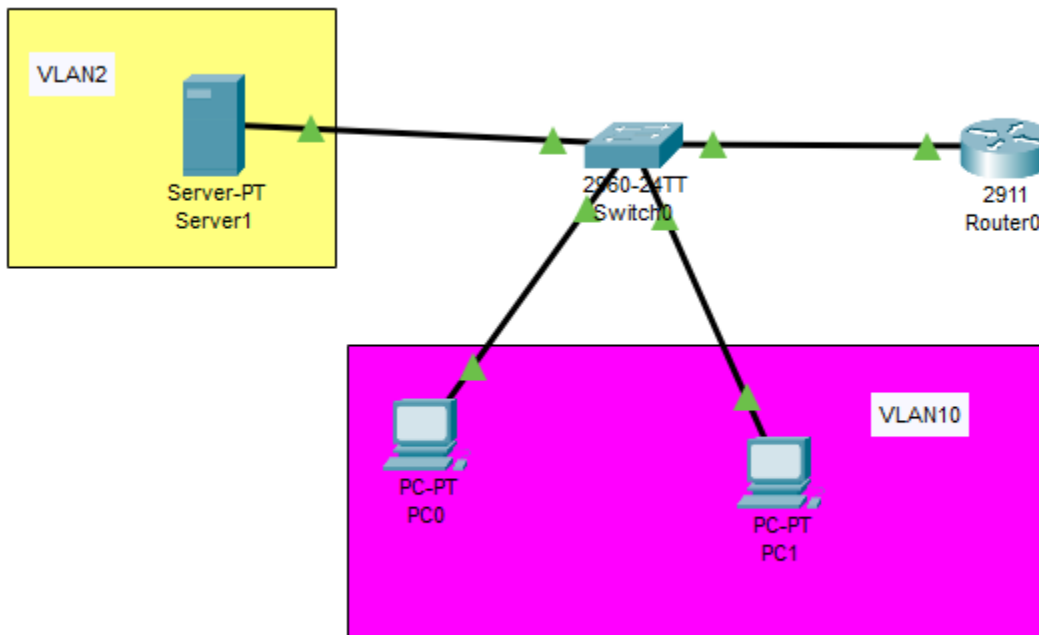
Steps:

- Set up a network using a server, router, switch and 2 PC's.
- Configure switches and as in VLANs.
- Configure router for inter VLAN routing.
- To implement ACL:
- Now, apply an ACL to restrict PC1 from accessing the Web Server on Port 80.
Router(config)# access-list 100 permit tcp host 192.168.10.10 any eq 80
Router(config)# access-list 100 deny tcp host 192.168.10.20 any eq 80
Router(config)# access-list 100 permit ip any any
here,
- PC0 (192.168.10.10) -> Allowed HTTP (Port 80) Traffic to Any Destination
- PC1 (192.168.10.20) -> Denied HTTP (Port 80) Traffic to Any Destination
- All Other Traffic is Allowed

Testing:

- From PC0, I tried accessing the Web Server on Port 80, the result was successful.
- From PC1, I tried accessing the Web Server on Port 80, host is not reachable indicating that ACL has blocked the PC1 from accessing the port 80.
- From PC1, I have checked if the ping is still working, and it provided successful replies.
- I checked the ACLs using :
show access-lists

Network setup:



Switch configuration:

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name VLAN10
Switch(config-vlan)#ex
Switch(config)#vlan 20
Switch(config-vlan)#name VLAN20
Switch(config-vlan)#ex
Switch(config)#interface FastEthernet0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#ex
Switch(config)#interface FastEthernet0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#ex
Switch(config)#interface FastEthernet0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#ex
Switch(config)#interface GigabitEthernet0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20
Switch(config-if)#ex
Switch(config)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
```

Router configuration and ACL configuration:

```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface GigabitEthernet0/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#interface GigabitEthernet0/0.10
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.10, changed state to up

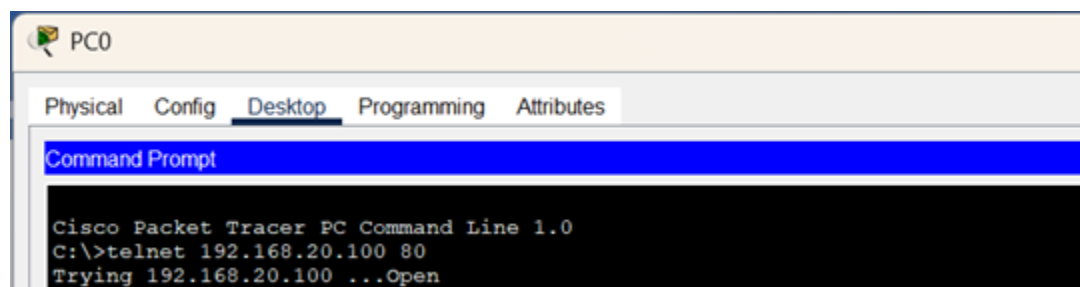
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.10, changed state to up

Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#ex
Router(config)#interface GigabitEthernet0/0.20
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.20, changed state to up

Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#ex
Router(config)#access-list 100 permit tcp host 192.168.10.10 any eq 80
Router(config)#access-list 100 deny tcp host 192.168.10.20 any eq 80
Router(config)#access-list 100 permit ip any any
Router(config)#interface GigabitEthernet0/0.10
Router(config-subif)#ip access-group 100 in
Router(config-subif)#exit
Router(config)#exit
Router#
```

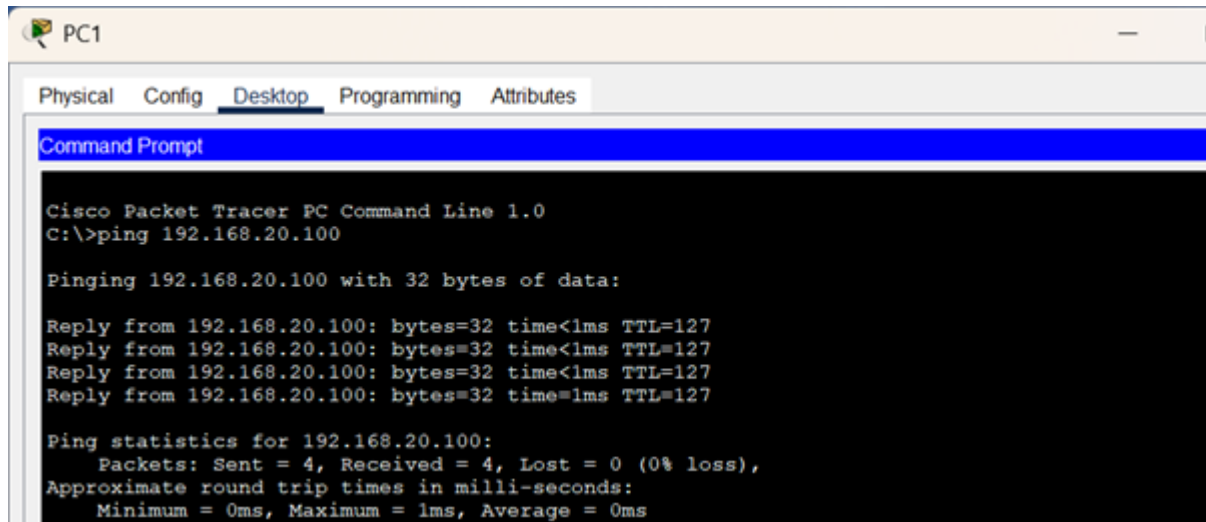
i) The server is accessible from PC0:



ii) Unable to access the server from PC1:

```
C:\>
C:\>telnet 192.168.20.100 80
Trying 192.168.20.100 ...
% Connection timed out; remote host not responding
```

PC1 can still ping the server:



The screenshot shows a PC1 window with the 'Desktop' tab selected. Inside is a 'Command Prompt' window titled 'Cisco Packet Tracer PC Command Line 1.0'. The command 'C:\>ping 192.168.20.100' has been entered. The output shows four successful replies from 192.168.20.100, each with 32 bytes of data, a time of less than 1ms, and a TTL of 127. The ping statistics show 4 packets sent, 4 received, and 0% loss, with a minimum round trip time of 0ms, a maximum of 1ms, and an average of 0ms.

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.20.100

Pinging 192.168.20.100 with 32 bytes of data:

Reply from 192.168.20.100: bytes=32 time<1ms TTL=127
Reply from 192.168.20.100: bytes=32 time<1ms TTL=127
Reply from 192.168.20.100: bytes=32 time<1ms TTL=127
Reply from 192.168.20.100: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.20.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Access table:

```
Router>enable
Router#show access-lists
Extended IP access list 100
  10 permit tcp host 192.168.10.10 any eq www (4 match(es))
  20 deny tcp host 192.168.10.20 any eq www (12 match(es))
  30 permit ip any any (7 match(es))
```

Q11. Configure a standard Access Control List (ACL) on a router to permit traffic from a specific IP range. Test connectivity to verify the ACL is working as intended.

I have set up a Standard ACL to permit traffic from a specific IP range while blocking all other traffic using the same network setup used in Question 10.

Steps:

- IP Range to Allow: 192.168.10.0/24 (all devices in VLAN 10)
- Deny All Other Traffic
- Apply ACL to the Interface for Incoming Traffic
- To achieve the router is configured with :
Router(config)# access-list 10 permit 192.168.10.0 0.0.0.255 (Allows all devices in VLAN 10 (192.168.10.x))
Router(config)# access-list 10 deny any (denies others)

Observations:

I used the ping command to check the connectivity on PC0 and PC1 and got successful replies. But i tried to use telnet and ssh on the server but it failed because we have blocked VLAN20 devices from accessing the VLAN10's service.

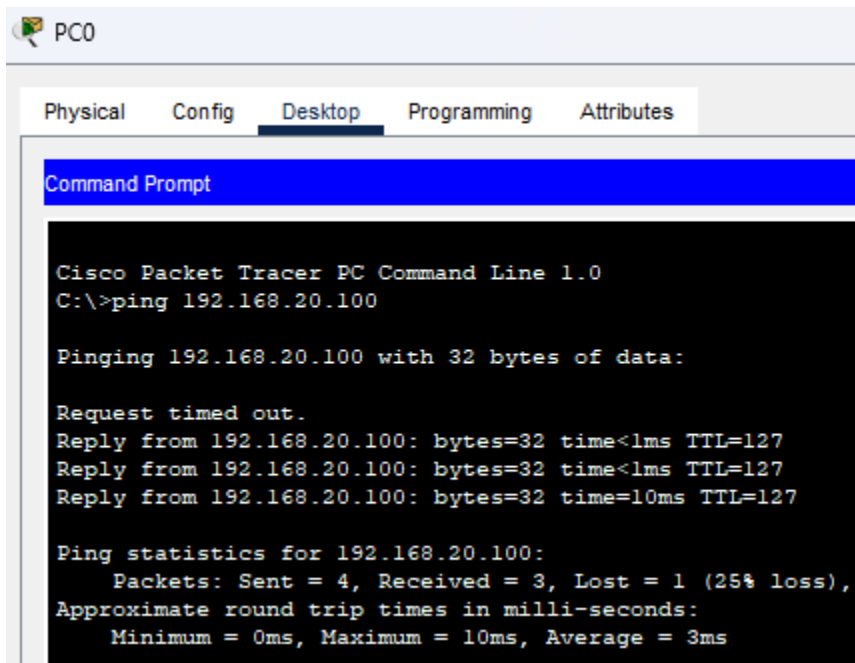
Configuring router:

```
Router(config)#access-list 10 permit 192.168.10.0 0.0.0.255
Router(config)#access-list 10 deny any
Router(config)#interface GigabitEthernet0/0.10
Router(config-subif)#ip access-group 10 in
Router(config-subif)#ex
Router(config)#ex
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show access-lists
Extended IP access list 100
  10 permit tcp host 192.168.10.10 any eq www
  20 deny tcp host 192.168.10.20 any eq www
  30 permit ip any any
Standard IP access list 10
  10 permit 192.168.10.0 0.0.0.255
  20 deny any

Router#show ip interface GigabitEthernet0/0.10
GigabitEthernet0/0.10 is up, line protocol is up (connected)
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound access list is 10
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
  IP fast switching is disabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP Fast switching turbo vector
```

To check connectivity on PC0 (192.168.10.10):



The screenshot shows the PC0 interface in Cisco Packet Tracer. The 'Desktop' tab is selected, and the 'Command Prompt' window is open. The command prompt displays the output of a ping command from 192.168.10.10 to 192.168.20.100. The output shows a 25% loss of packets.

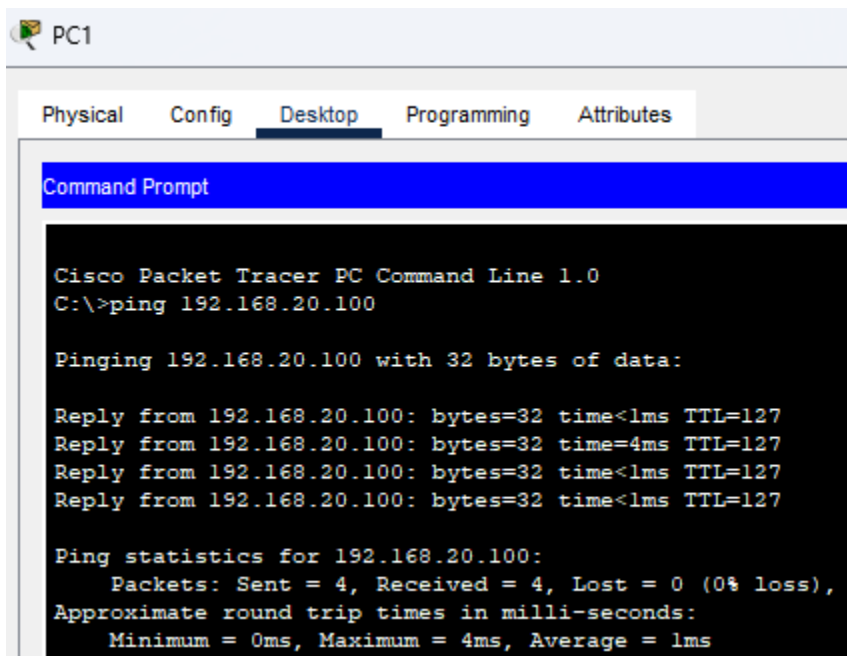
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.20.100

Pinging 192.168.20.100 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.100: bytes=32 time<1ms TTL=127
Reply from 192.168.20.100: bytes=32 time<1ms TTL=127
Reply from 192.168.20.100: bytes=32 time=10ms TTL=127

Ping statistics for 192.168.20.100:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 3ms
```

To check connectivity on PC1 (192.168.10.20):



The screenshot shows the PC1 interface in Cisco Packet Tracer. The 'Desktop' tab is selected, and the 'Command Prompt' window is open. The command prompt displays the output of a ping command from 192.168.10.20 to 192.168.20.100. The output shows 0% loss of packets.

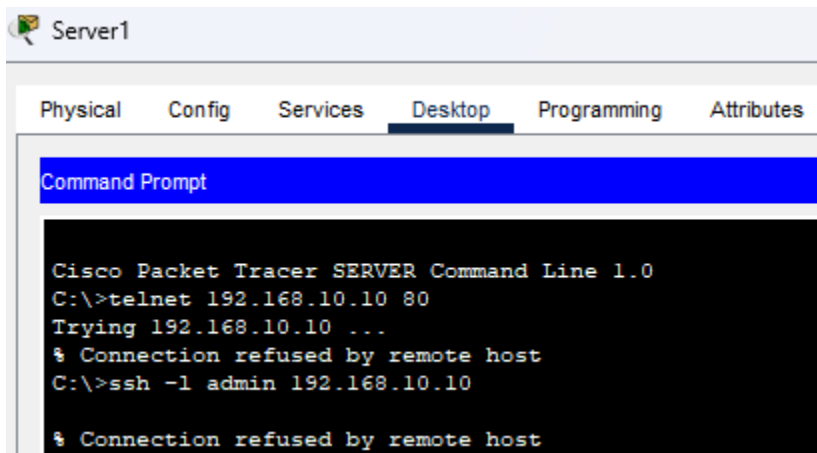
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.20.100

Pinging 192.168.20.100 with 32 bytes of data:

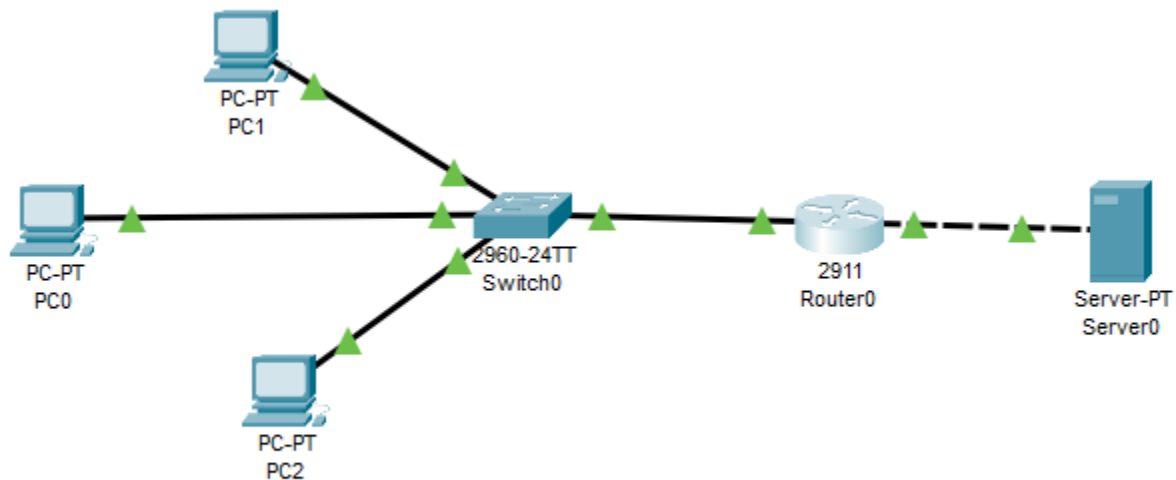
Reply from 192.168.20.100: bytes=32 time<1ms TTL=127
Reply from 192.168.20.100: bytes=32 time=4ms TTL=127
Reply from 192.168.20.100: bytes=32 time<1ms TTL=127
Reply from 192.168.20.100: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.20.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 4ms, Average = 1ms
```

Test Command from Server (192.168.20.100) to PC0 (192.168.10.10) - telnet and ssh:



Q13. Try Static NAT, Dynamic NAT and PAT to translate IPs



```
Router(config-if)#access-list 20 permit 10.0.0.0 0.0.0.255
Router(config)#ip nat pool cn 20.0.0.1 20.0.0.1 netmask 255.0.0.0
Router(config)#ip nat inside source list 20 pool cn overload
Router(config)#interface GigabitEthernet 0/0
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#interface GigabitEthernet 0/1
Router(config-if)#ip nat outside
Router(config-if)#exit
```



```
Router#show ip nat translation
Pro  Inside global      Inside local      Outside local      Outside global
icmp 20.0.0.1:1        10.0.0.1:1        50.0.0.2:1        50.0.0.2:1
icmp 20.0.0.1:2        10.0.0.1:2        50.0.0.2:2        50.0.0.2:2
icmp 20.0.0.1:3        10.0.0.1:3        50.0.0.2:3        50.0.0.2:3
icmp 20.0.0.1:4        10.0.0.1:4        50.0.0.2:4        50.0.0.2:4
icmp 20.0.0.1:5        10.0.0.1:5        50.0.0.2:5        50.0.0.2:5
icmp 20.0.0.1:6        10.0.0.1:6        50.0.0.2:6        50.0.0.2:6
icmp 20.0.0.1:7        10.0.0.1:7        50.0.0.2:7        50.0.0.2:7
icmp 20.0.0.1:8        10.0.0.1:8        50.0.0.2:8        50.0.0.2:8
```

Q14. Download iPerf in laptop/phone and make sure they are in same network. Try different iPerf commands with tcp, udp, bidirectional, reverse, multicast, parallel options and analyze the bandwidth and rate of transmission, delay, jitter etc.

iPerf is a network testing tool used to measure bandwidth and performance. It operates in a client-server model, supporting TCP, UDP, multicast, and reverse testing. It measures throughput, latency, jitter, and packet loss.

1. Basic Client-Server TCP Test

Command for Server: `iperf3 -s`

Explanation: The server listens for incoming connections and measures the performance.

Command for Client: `iperf3 -c <server_ip>`

Explanation: The client connects to the server and sends TCP traffic to measure throughput.

2. UDP Test

Command for Server: `iperf3 -s`

Explanation: The server listens for UDP traffic.

Command for Client: `iperf3 -c <server_ip> -u -b 100M`

Explanation: The client sends UDP traffic to the server with a bandwidth of 100 Mbps. `-u` specifies UDP mode, and `-b` sets the bandwidth.

3. Test with Specific Port

Command for Server: `iperf3 -s -p 12345`

Explanation: The server listens on port 12345.

Command for Client: `iperf3 -c <server_ip> -p 12345`

Explanation: The client connects to the server on port 12345.

4. Reverse Mode Test

Command for Server: `iperf3 -s`

Explanation: The server listens for incoming connections.

Command for Client: `iperf3 -c <server_ip> -R`

Explanation: The -R flag reverses the data flow, meaning the server sends data to the client instead of the client sending data to the server. Useful for testing server upload capacity.

5. Bidirectional (Duplex) Test

Command for Server: `iperf3 -s`

Explanation: The server listens for connections.

Command for Client: `iperf3 -c <server_ip> -d`

Explanation: The -d flag enables bidirectional testing, where data flows in both directions simultaneously. Measures throughput in both directions (client-to-server and server-to-client).