CYBER SECURITY UPSKILLING PROGRAM

قدم خلال مبادرة زنك/2 في جامعة البلقاء التطبيقية
بالتعاون مع أكاديمية سايبر شيلد

AUG 2024

Networks Part

Version 1

INST.:ENG.ALI BANI BAKAR-0778642376(CYBER SHIELD ACADEMY)

DONE BY: ENG. Dana Al-Mahrouk-0798697842-BAU.UNIV.

Outline

1. Networks
2. Linux Essentials
3. Cybersecurity Foundation
4. Ethical Hacking
5. Digital Forensic Investigation

Networking

- TCP/IP & OSI Model
- Network Devices
- Host Devices
- Tool:
  - Packet Tracer
  - Wireshark
  - GNS3

Day 1

- Outline:
  1. TCP/IP Model
  2. Network Devices
     - Switch
     - Router
  3. Host Devices
     - PC
     - Server
       I. Web Server
       II. DNS Server
  4. Message & Ping
  5. Broadcast & ARP Table
  6. Project at packet tracer

TCP/IP Model

- Have 5 Layers
- From CISCO
- Encapsulation:

Each layer adds its own data [control data] to the [Data].

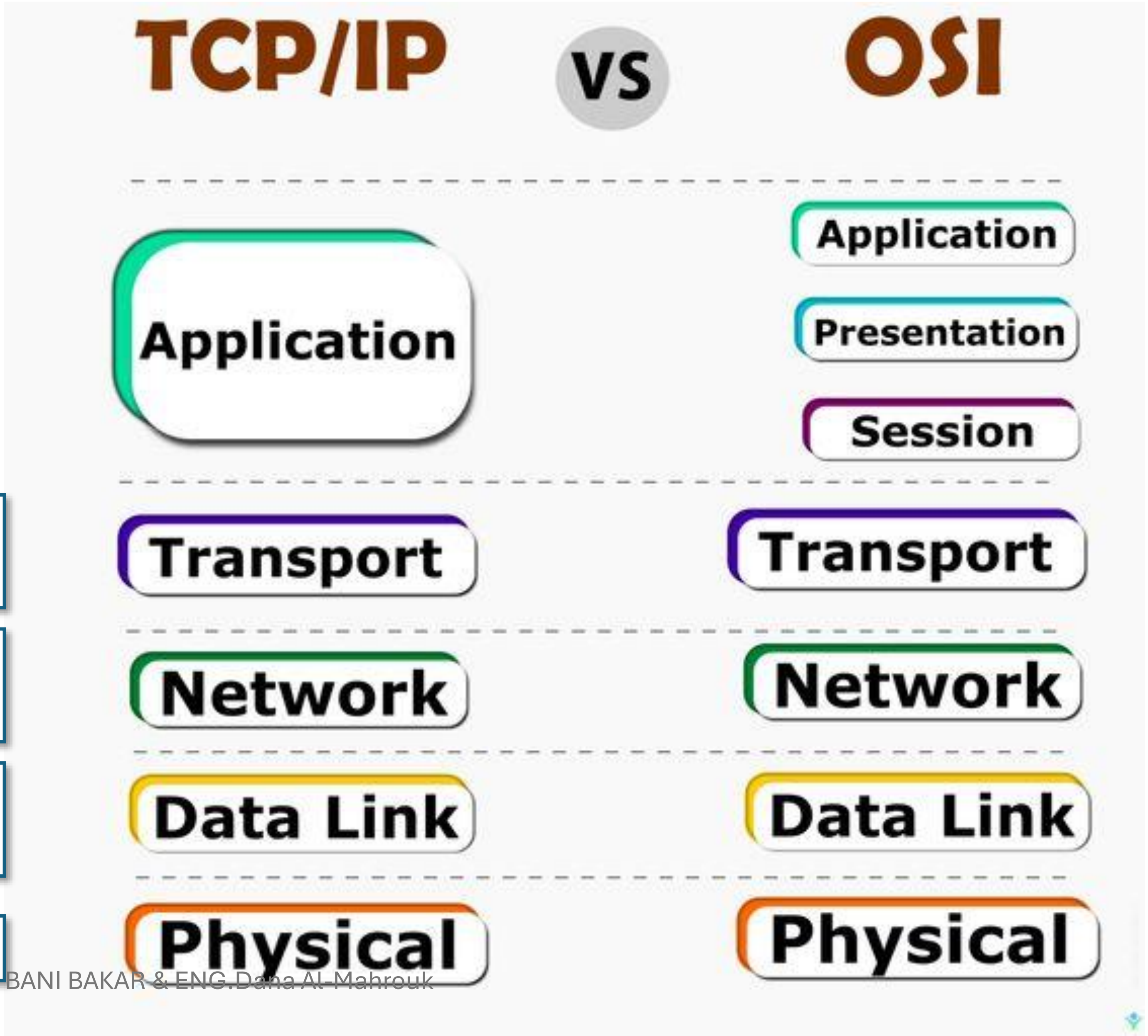
Data

[**Transport Header (Port #)** | Data] ➔ Segment

[**Network Header (IP)** | Port # | Data] ➔ Packet

[**Data Link Header (MAC)** | IP | Port # | Data] ➔ Frame

Bits



Network Devices

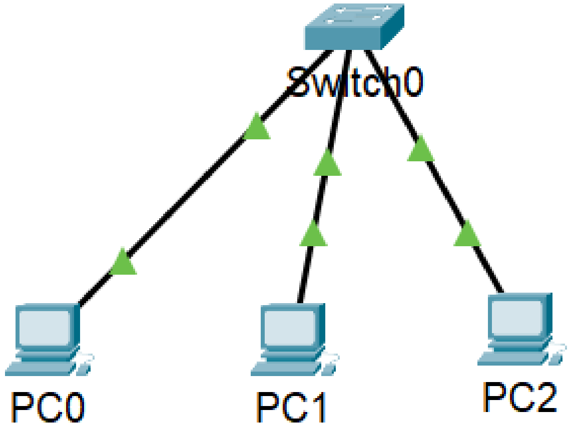
1. Switch:

Connects between two host devices (End devices), one Network.

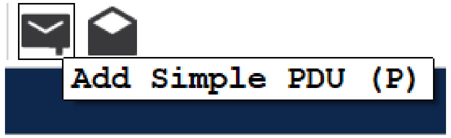
Work at layer 2 (Data Link Layer) ➔ MAC Address.

Switch have ports:

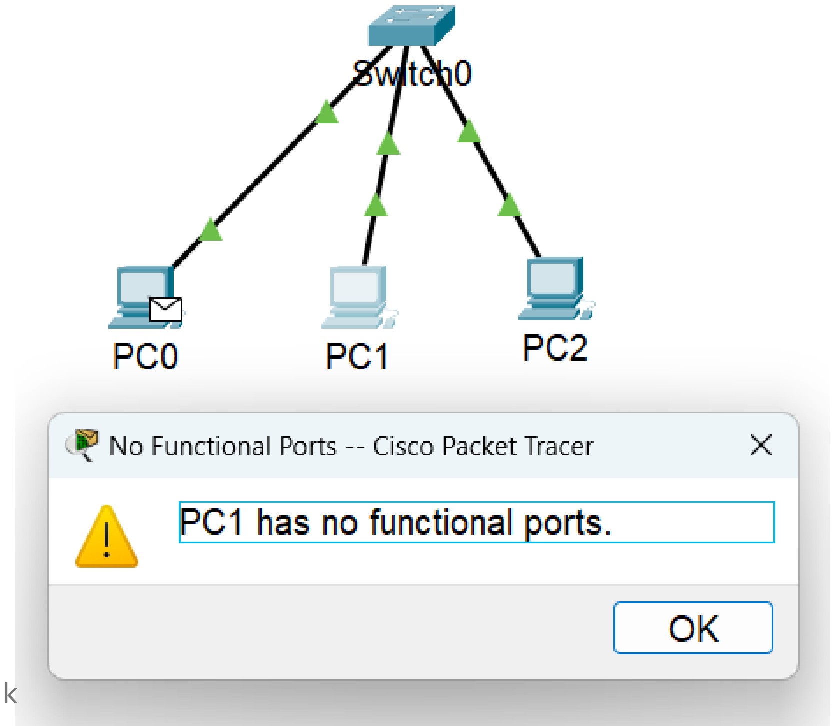
- Console
- Fast Ethernet [0/1 – 0/24 || 0/32]
- Gigabit Ethernet [0/1 – 0/2]



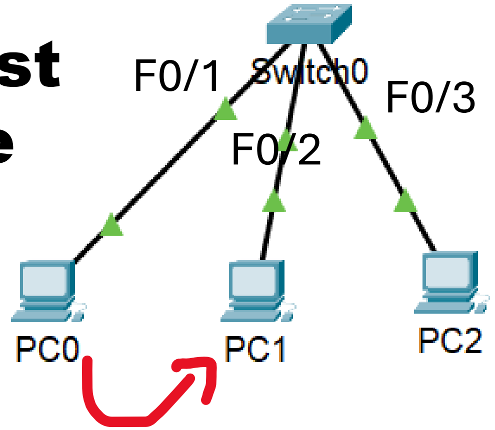
Message



- Test connection: (Ping) ➔ ICMP 'L3 Network '
  - Echo request
  - Echo replay / response
- Add message ➔ **Error** ➔ PC has no functional ports ➔ PC need IP Address
- **Solution** ➔ PC ➔ Desktop ➔ IP Configuration :
  - IPv4 Address: 192.168.0.2 ~ 192.168.0.4
  - Subnet Mask: 255.255.255.0 ➔ Default no change
- Q: Switch works at layer 2 (Data link), it can see the MAC Address only (can not see the IP Address), 'MAC Address is a unique number comes with the device when you bought it', the question is: Why does the connection succussed only when we put IP Address –layer 3 (Network)- to PC?



Broadcast message
ARP L2



1) Send message ➔ ping

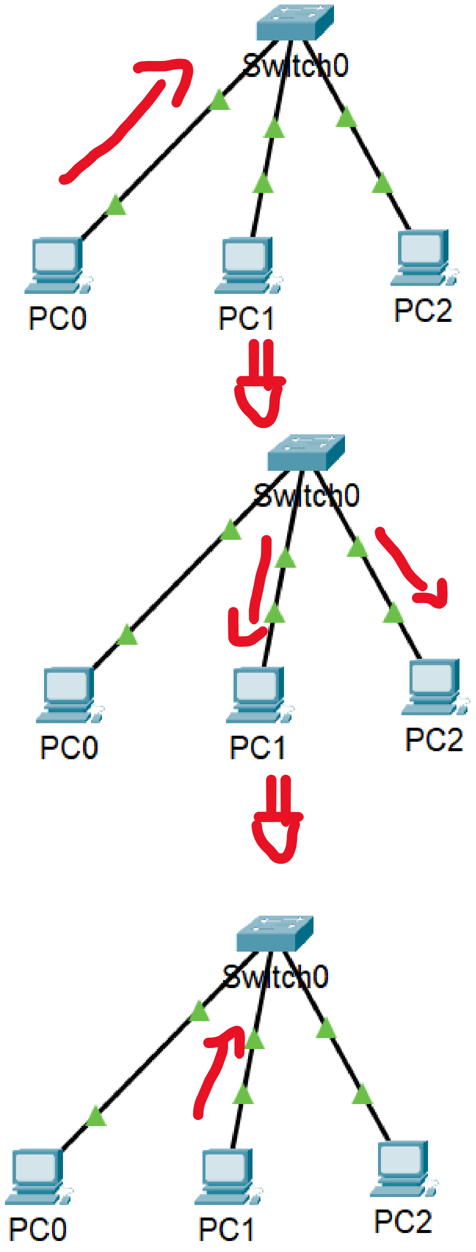
2)
Request:
[src MAC | dest MAC | src IP | dest IP | Data]
[02 | ?? | .0.2 | .0.3 | Data]

Sender does not know the MAC Address for Destination

Q: How can switch solve that?
A: By using **Broadcast**

Note:
Switch has a Table that contains **MAC | Port #**
➔ called MAC Table, Switch Table

MAC	Port #
02	FE 0/1
03	FE 0/2
04	FE 0/3



A) Ping <dest / target ip>

B) Broadcast MAC ➔ FF:FF:FF:FF:FF:FF
Switch sends the message to all devices (except sender), to identify the MAC Address of the Destination IP Address that sender needs to know.

Note:
Broadcast IP ➔ 255.255.255.255

C) Only the PC that has that dest IP will receive a message (that contain the IP Address for that PC) ➔ Uni-cast

Note:
Sender will save the IP MAC Address for that Receiver at ARP Table

arp -a

Replay:
[src MAC | dest MAC | src IP | dest IP | Data]
[03 | 02 | .0.3 | .0.2 | Data]

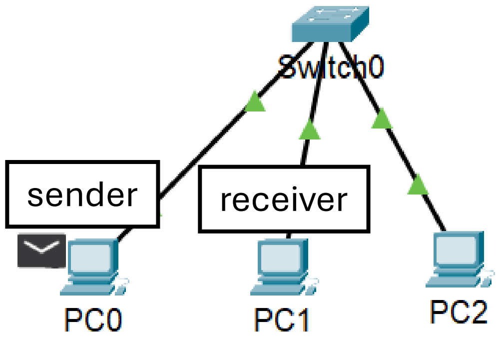
Ping message
ICMP L3

A) The sender sends the message to the switch, after searching for the MAC address and extracting it from the **ARP Table for PC0**.

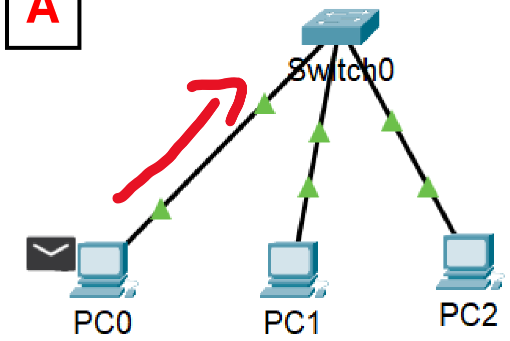
IP	MAC
.0.2	03
.0.3	04

B) The switch works on the L2, Data Link Layer, which depends on the MAC address of the sender and receiver.
The switch cannot see the layers above the second layer, so if the sender does not have the recipient's MAC address, the switch sends a Broadcast message.

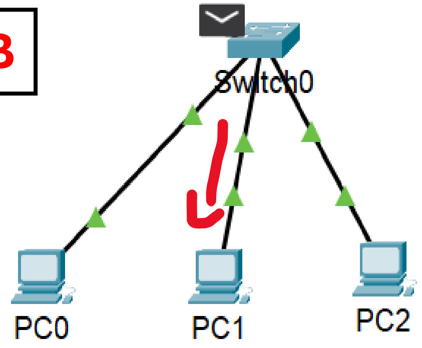
Echo Request



A

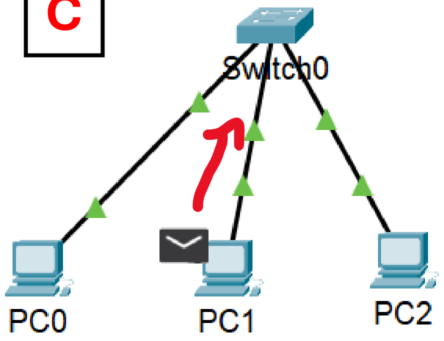


B



Echo Reply

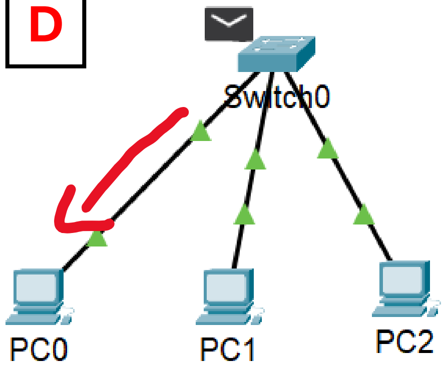
C



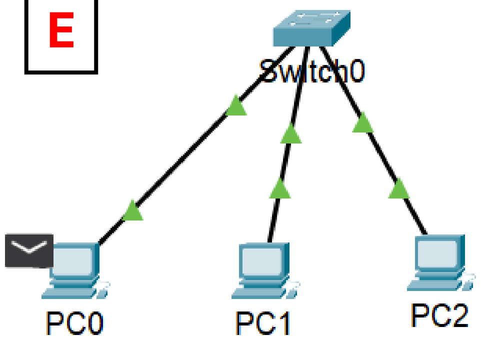
C) The message reaches the destination successfully; The receiver sends a **replay message** to the sender to confirm that the message has arrived successfully.

- The computer can deal with all network layers, so the computer receiving the message can open the message and **read the data**.
- If there is no computer with the IP address to which the sender wanted to send the message, it will be returned **Request Time** Out from switch.

D



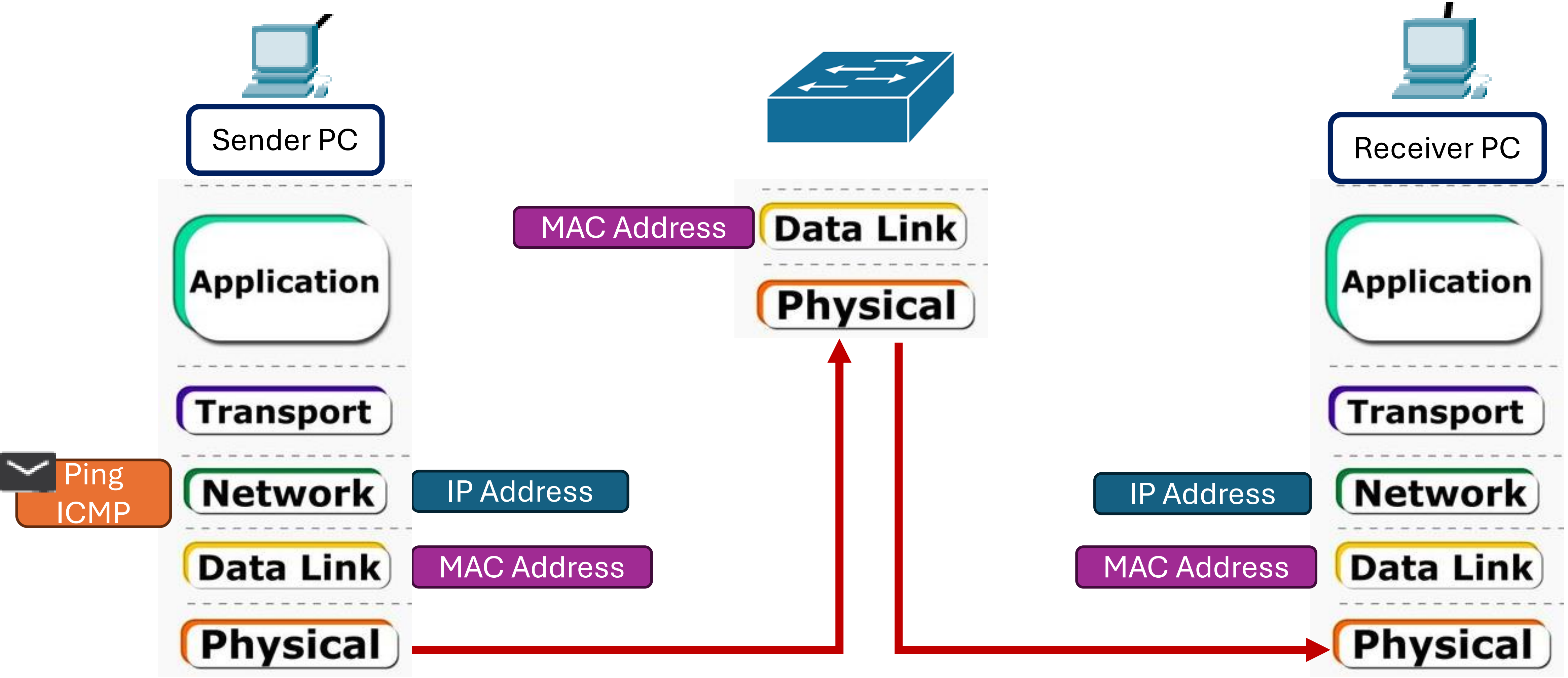
E



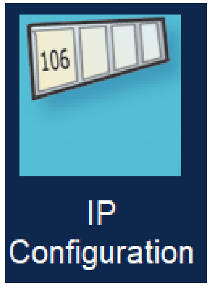
D & E) The switch sends this message to the sender, and it reaches the computer successfully

Q: ICMP use TCP or UDP protocol?
Not one of them. ICMP is a protocol in L3, the network layer, and TCP and UDP are in L4, the transport layer. Protocols in a specific layer use protocols in the layer below them.

Message Transport



Desktop

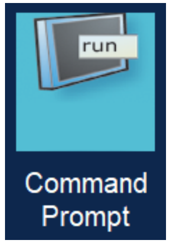


IPv4 Address	192.168.0.2
Subnet Mask	255.255.255.0

IPv4 Address	192.168.0.3
Subnet Mask	255.255.255.0

IPv4 Address	192.168.0.4
Subnet Mask	255.255.255.0

Desktop




```
C:\>ipconfig

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: FE80::2D0:D3FF:FE6E:8E1
   IPv6 Address....................: ::
   IPv4 Address....................: 192.168.0.2
   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:\>arp -a
No ARP Entries Found
```

```
C:\>ping 192.168.0.3

Pinging 192.168.0.3 with 32 bytes of data:

Reply from 192.168.0.3: bytes=32 time=38ms TTL=128
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 38ms, Average = 9ms
```

```
C:\>arp -a
  Internet Address      Physical Address      Type
  192.168.0.3           00d0.ff8d.ba97        dynamic
```

```
C:\>ping 192.168.0.5

Pinging 192.168.0.5 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

*] At the beginning, there is no ARP Table, however, when PC ping starts to send messages to other PCs at first time ➔ Broadcast sends to all PCs ➔ Target PC receives it own MAC Address ➔ Now, the sender PC get the MAC Address to Distance PC ➔ it saves it at ARP Table & after that it sends message.

**] If target PC found ➔ message [Broadcast message] + 4 echo request – replay ➔ (4 * 2) + 2 = 10

If not found ➔ Request Time out ➔ [Broadcast message] ➔ 2

Note:
ARP scanner ➔ fast and danger [Python]

[192.168.0.2 192.168.0.3 192.168.0.4]

Same Network 1

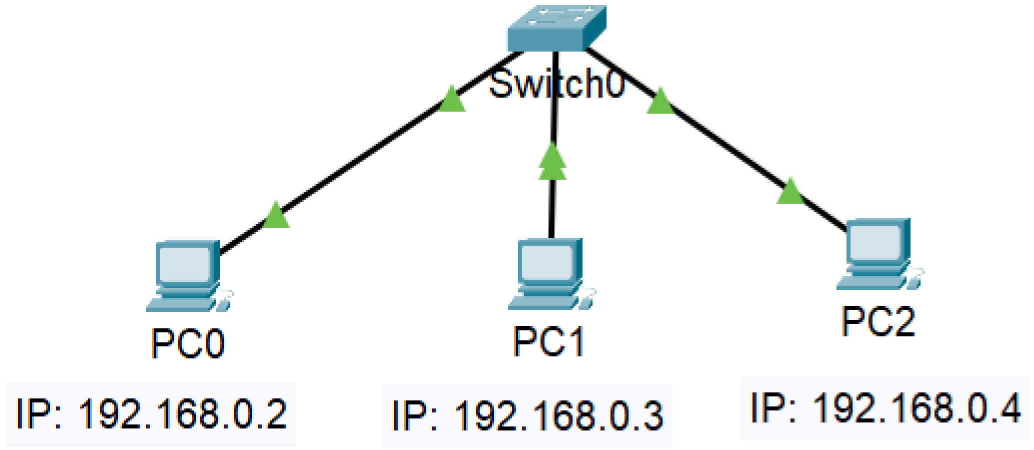
[192.168.1.2]

Network 2

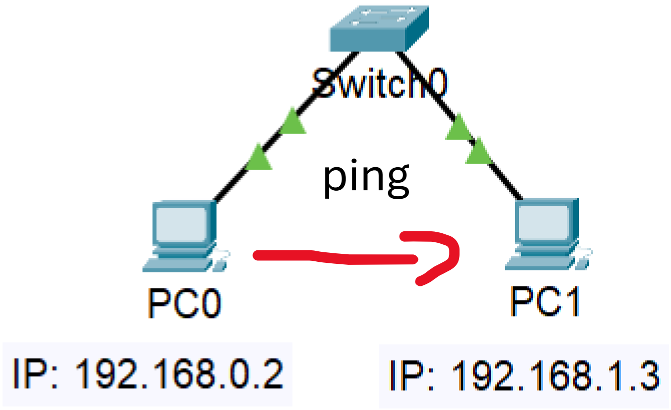
Consider subnet mask: 255.255.255.0

Switch can connect devices that can communicate in one network

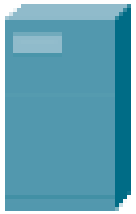
Connection success



Connection fail



Server



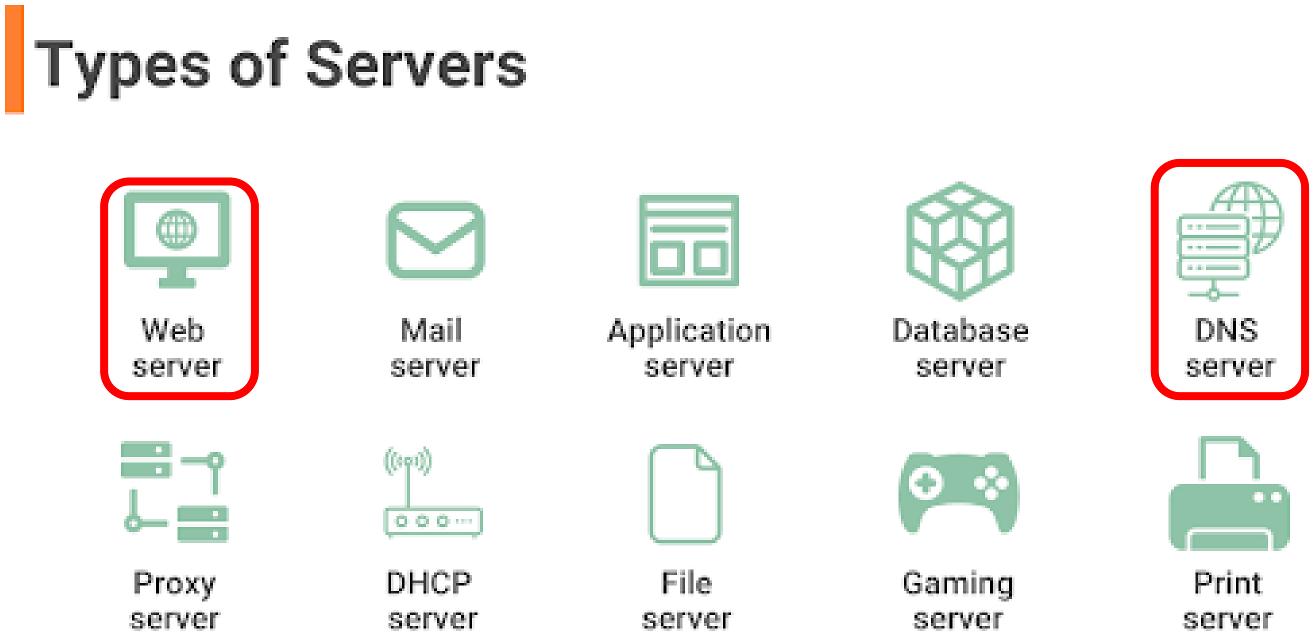
- It provides service to users using specific port.

- **Web Server:**

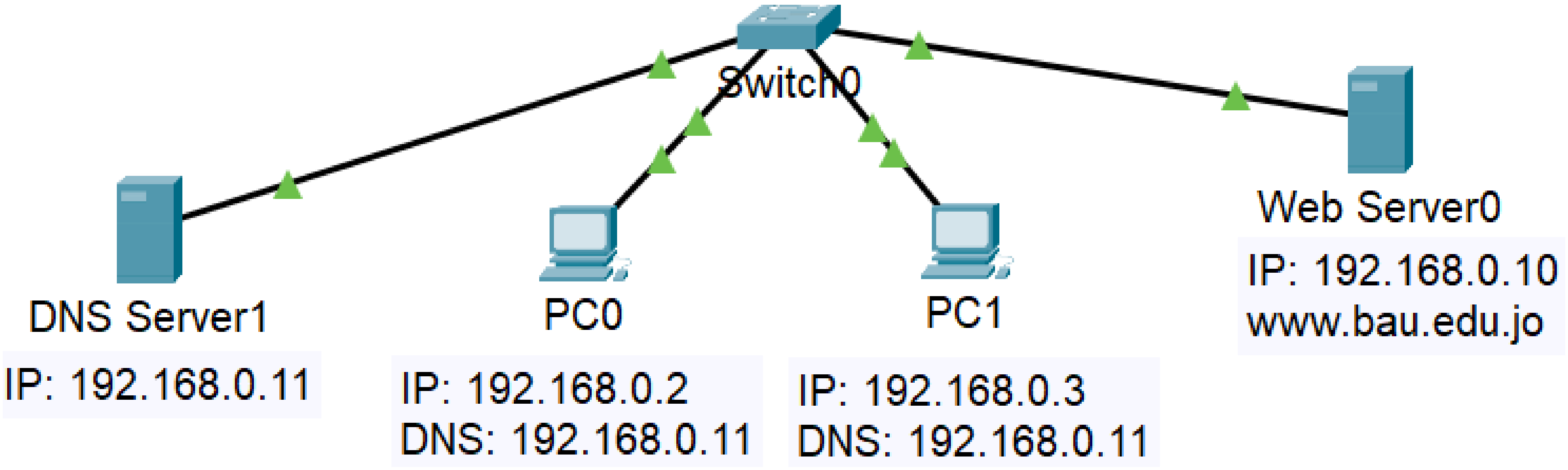
is a computer system capable of delivering web content to end users over the internet or intranet via a web browser

- **DNS server:**

translates domain names into IP addresses, enabling users to access websites using names.

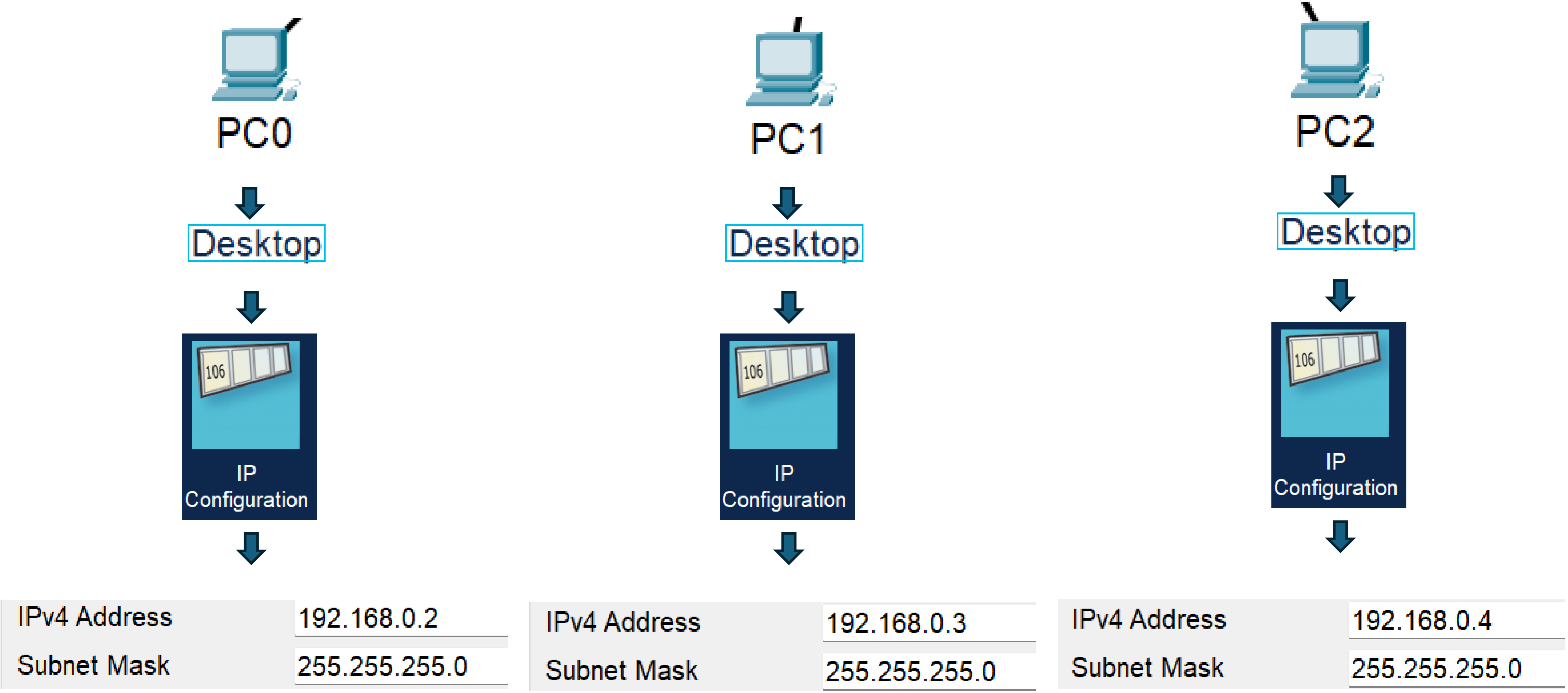


Packet Tracer Typology (CISCO)

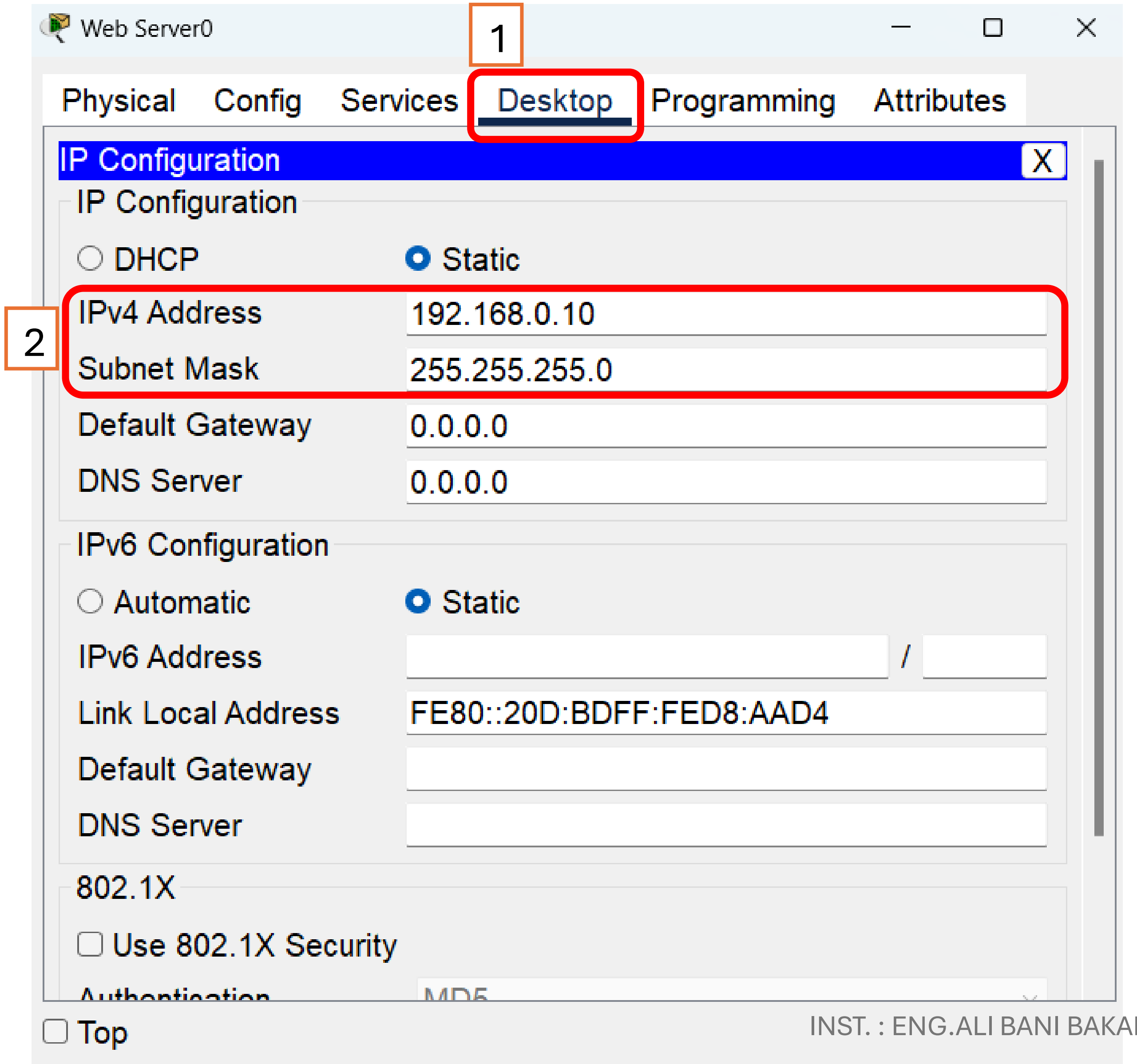


A. Give PCs IP Address.
B. Give Web Server IP Address
C. Give Web Server HTTP Service & write simple web page.
D. PC Connect with Web Server by IP Address
E. Give DNS Server IP Address & DNS Service, give Web Server a name.
F. Give PCs DNS Server & connect with Web Server by name

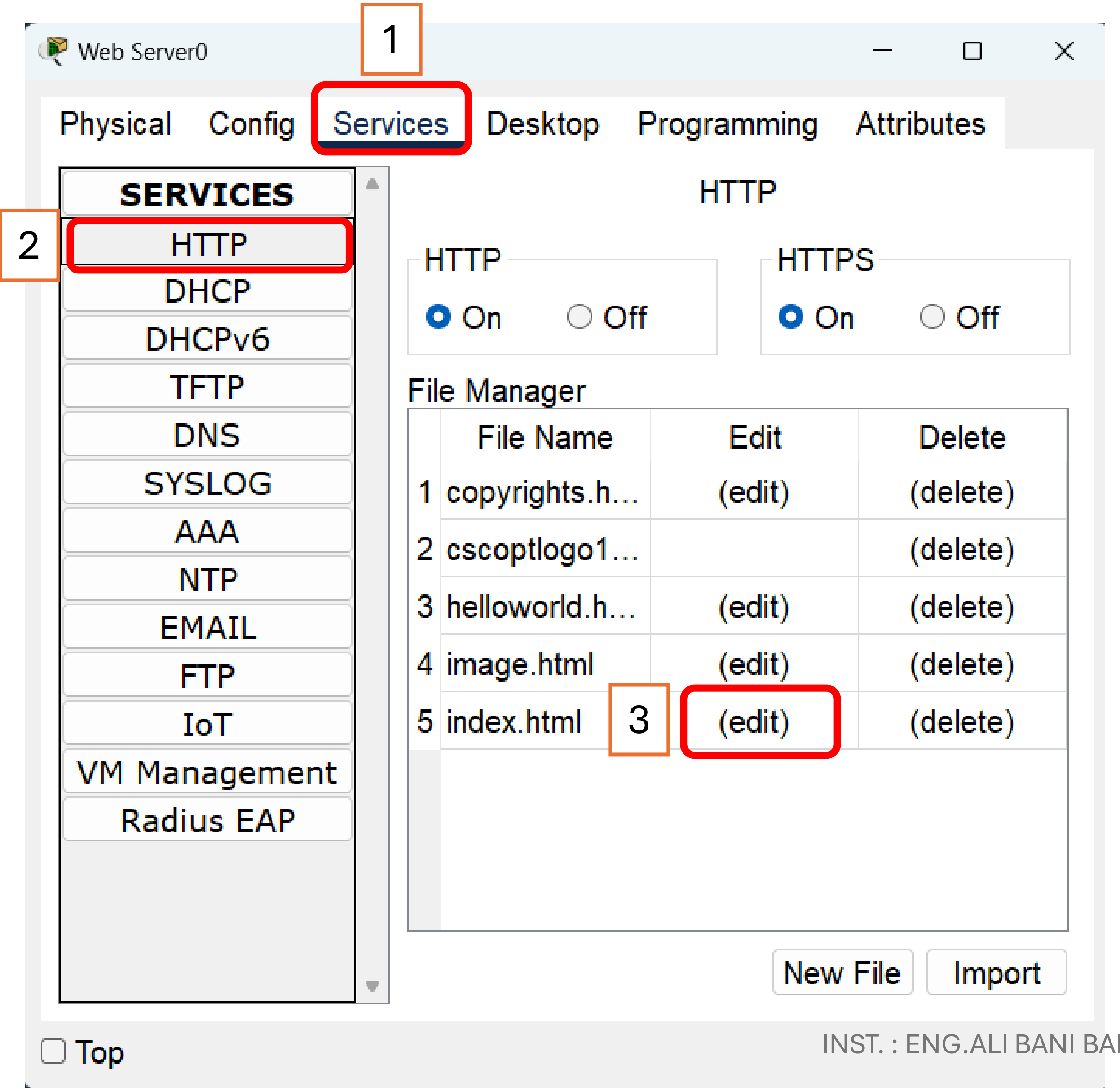
A) Give PCs IP Address

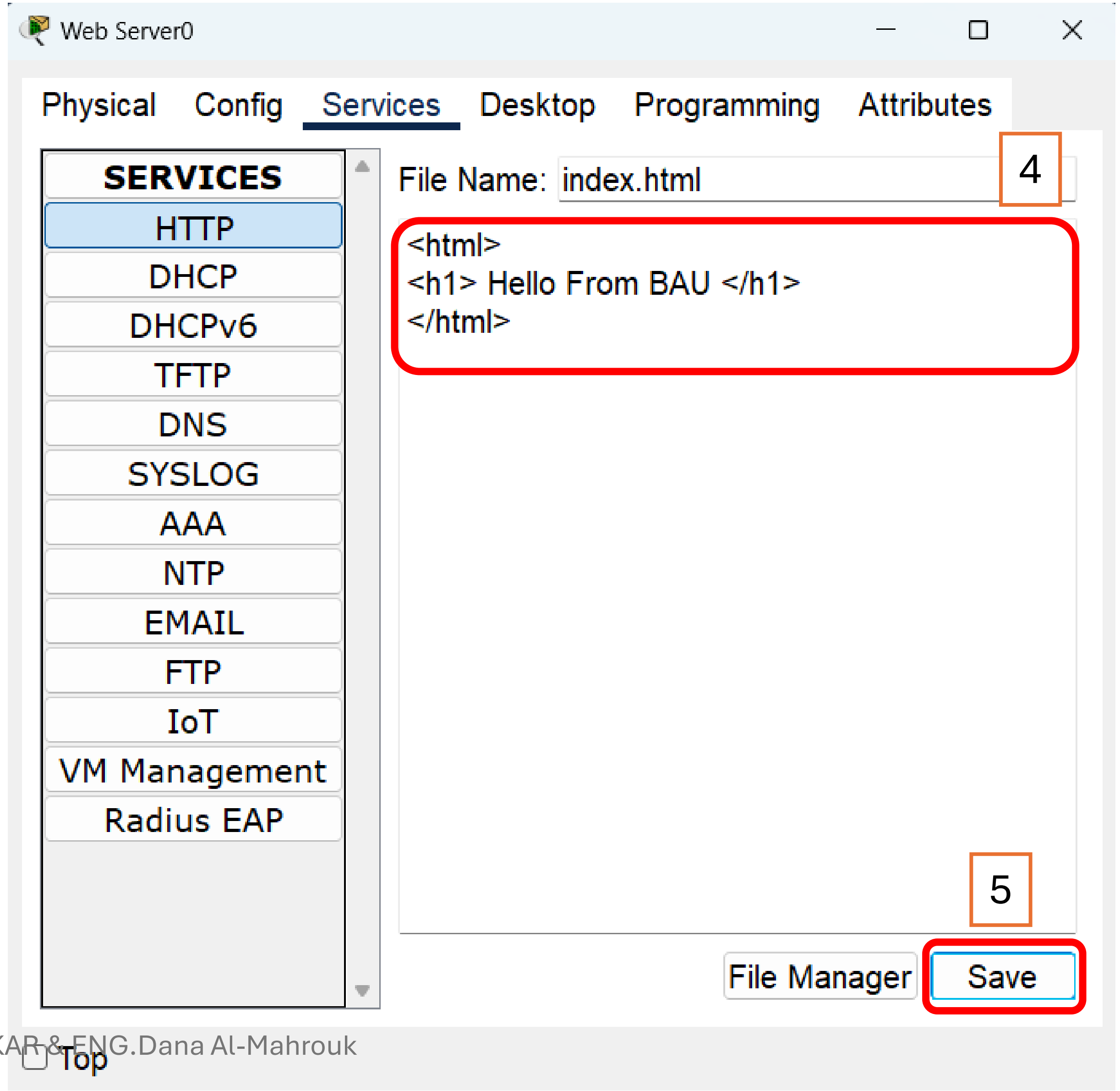


B) Give Web Server IP Address

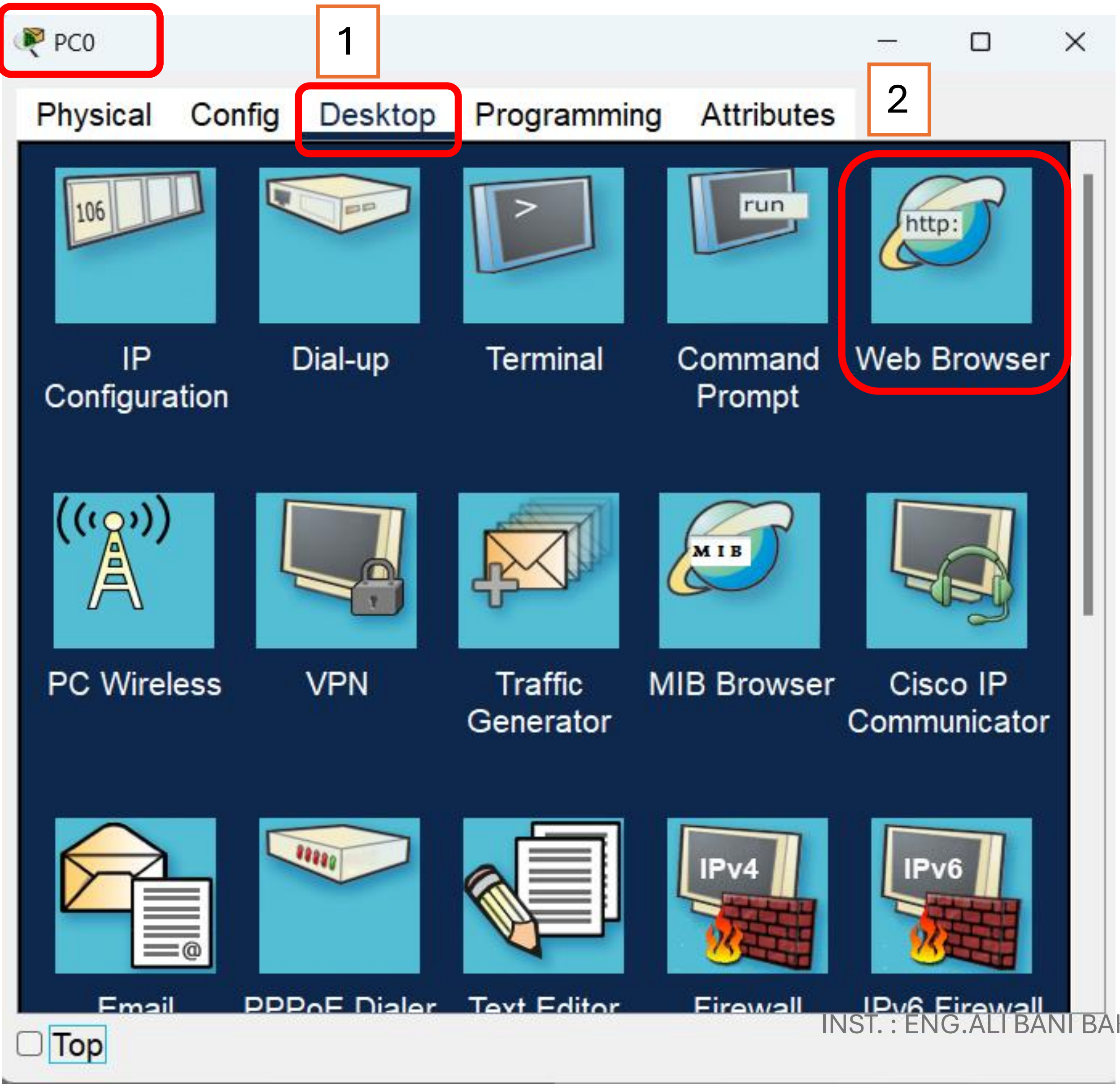


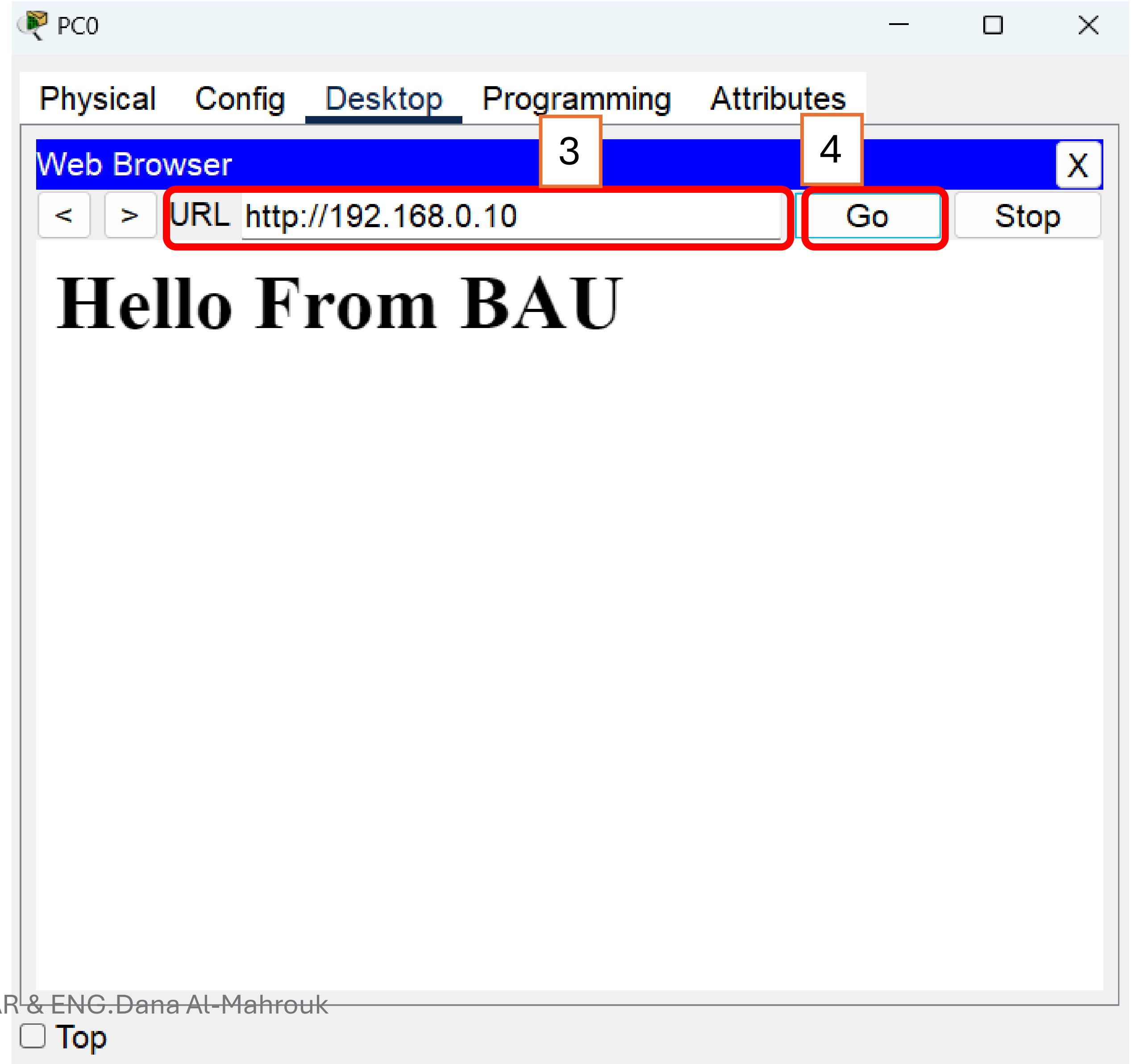
C) Give Web Server HTTP Service & write simple web page.





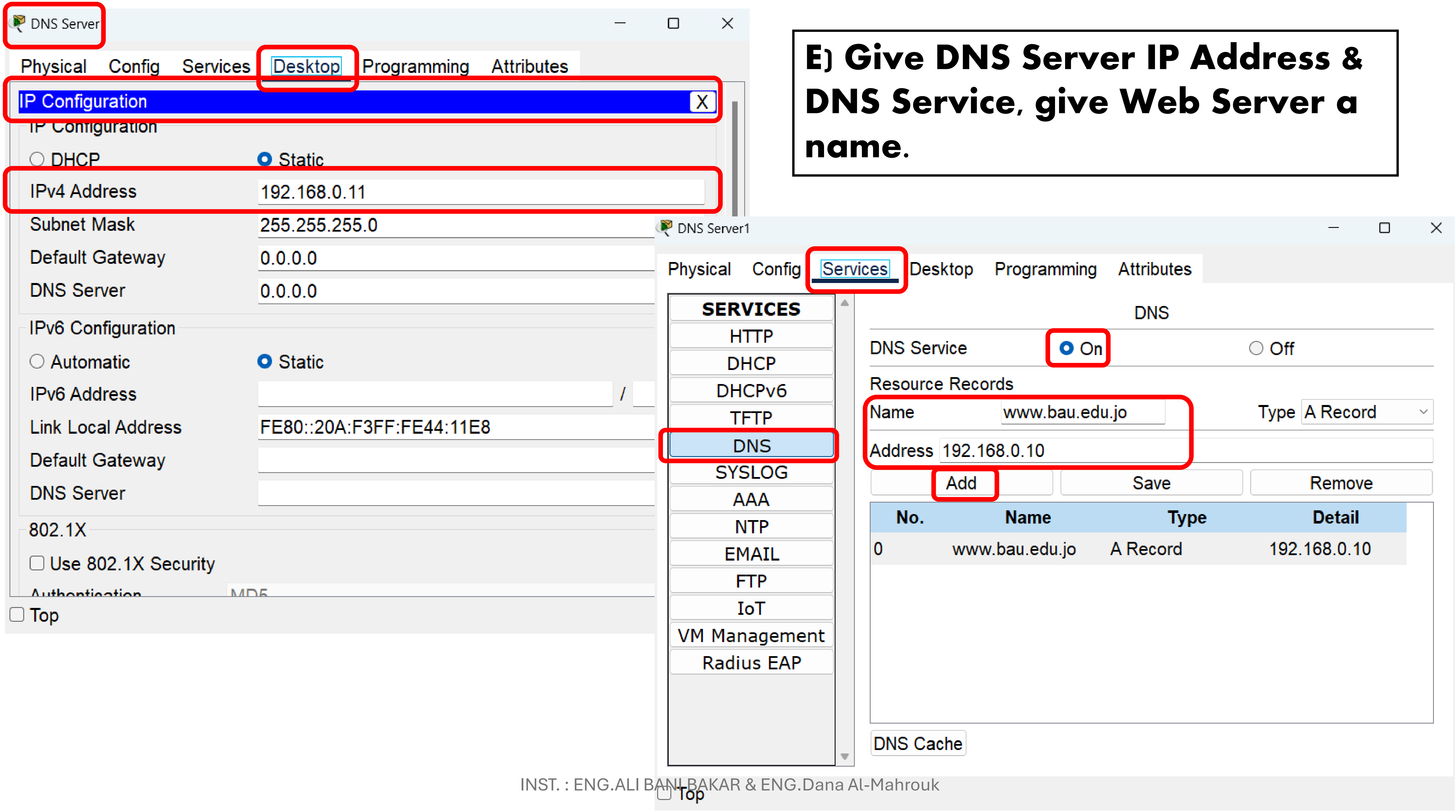
D) PC Connect with Web Server by IP Address



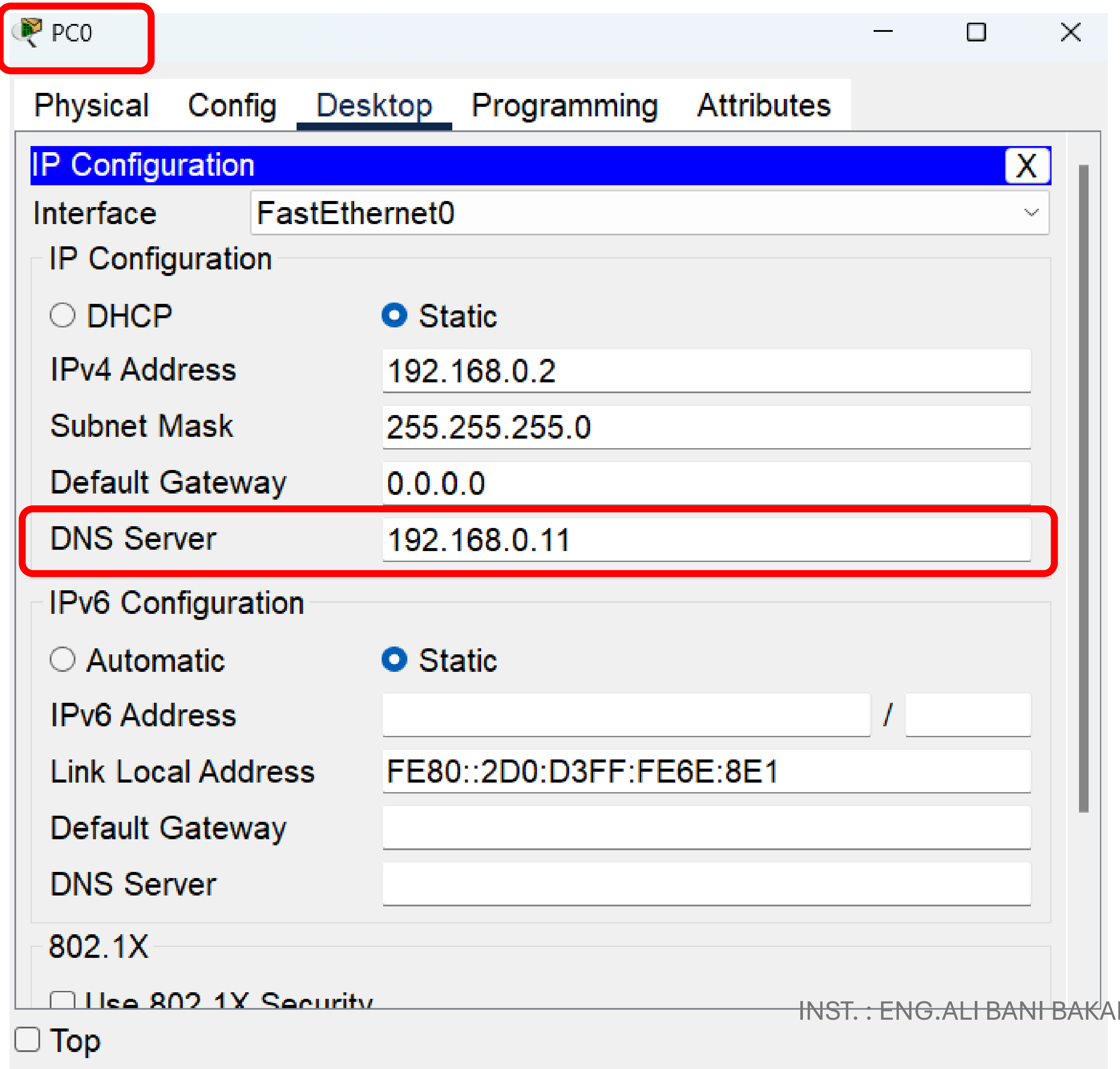


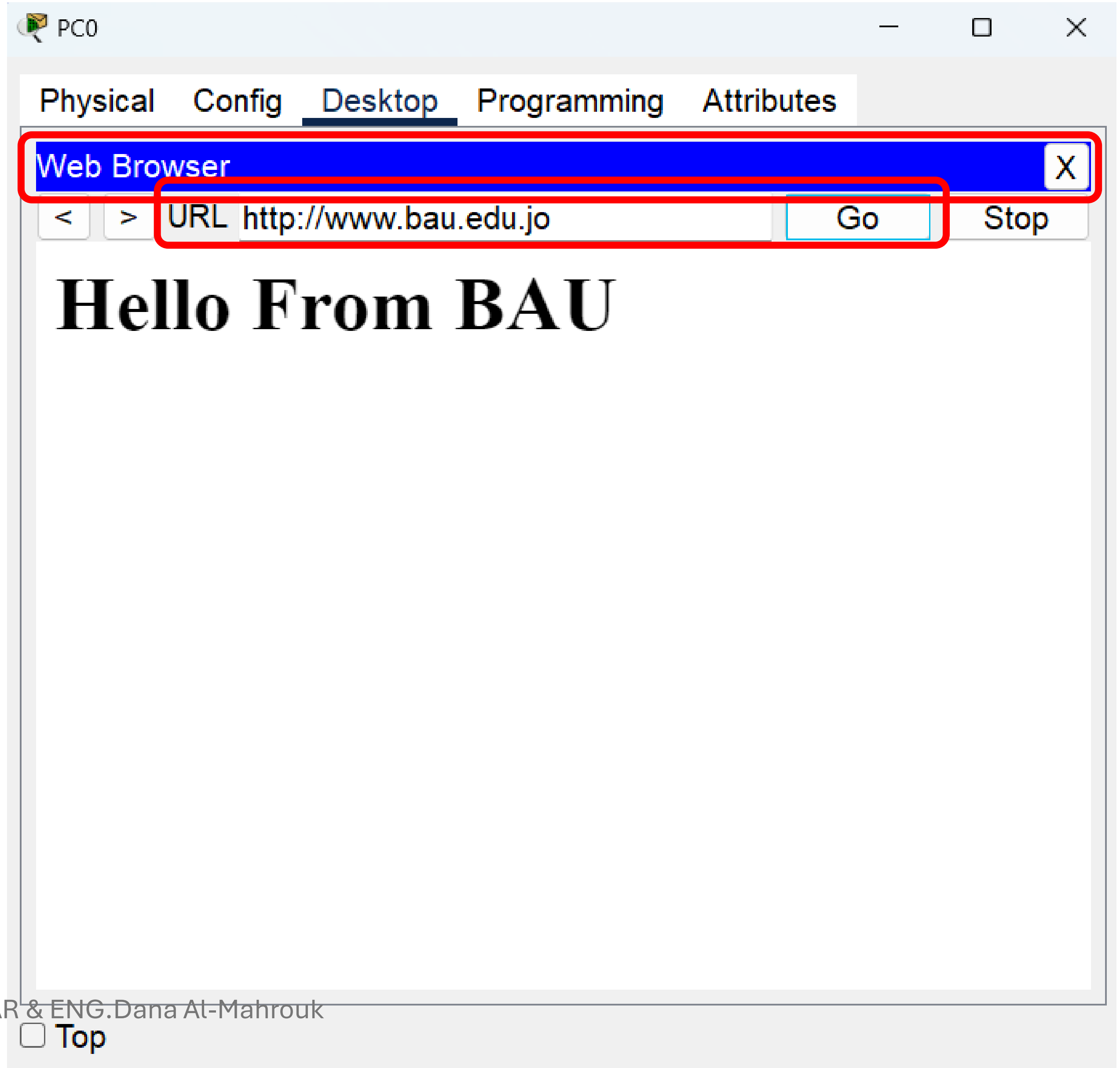
- Q: how to connect to a web page using DNS?
- A: Using DNS Server that convert name to IP Address of Web Server.

E) Give DNS Server IP Address & DNS Service, give Web Server a name.

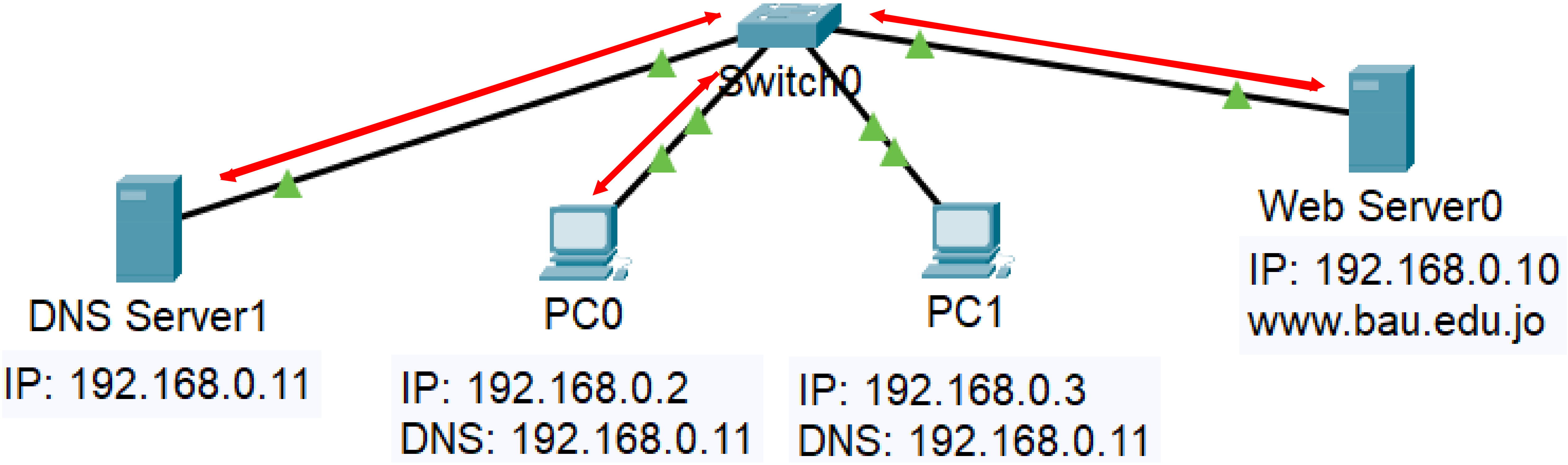


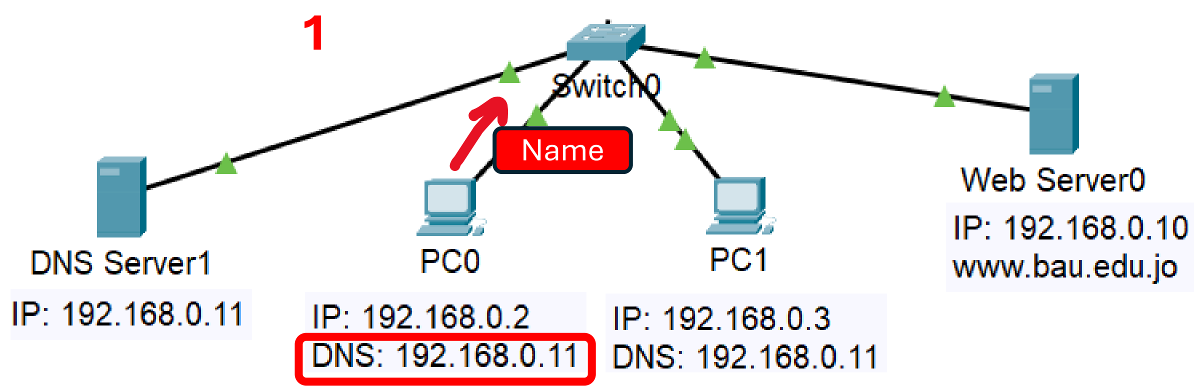
F) Give PCs DNS Server & connect with Web Server by name

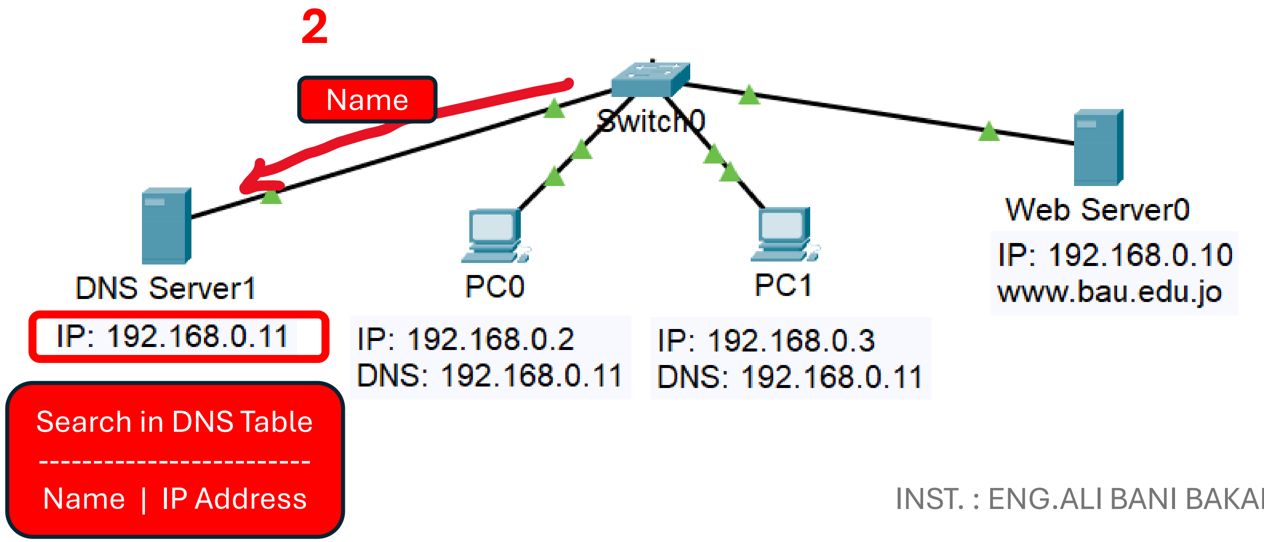




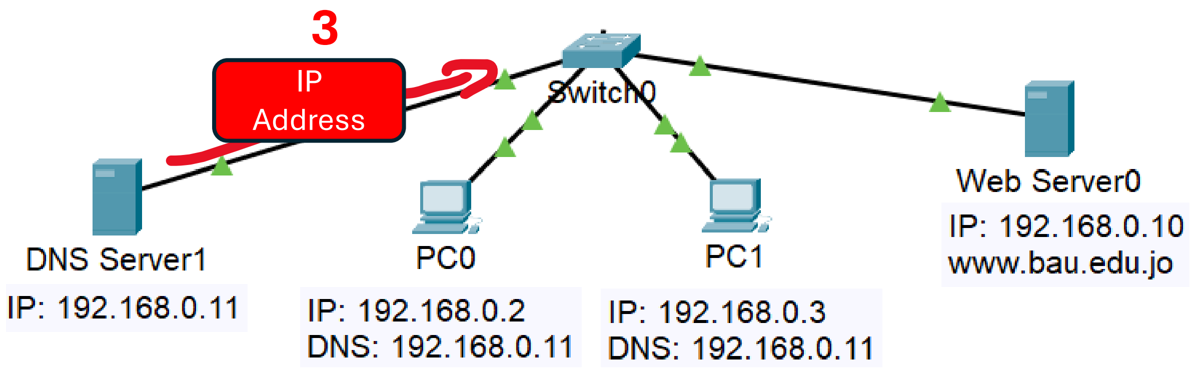
How It Work now?

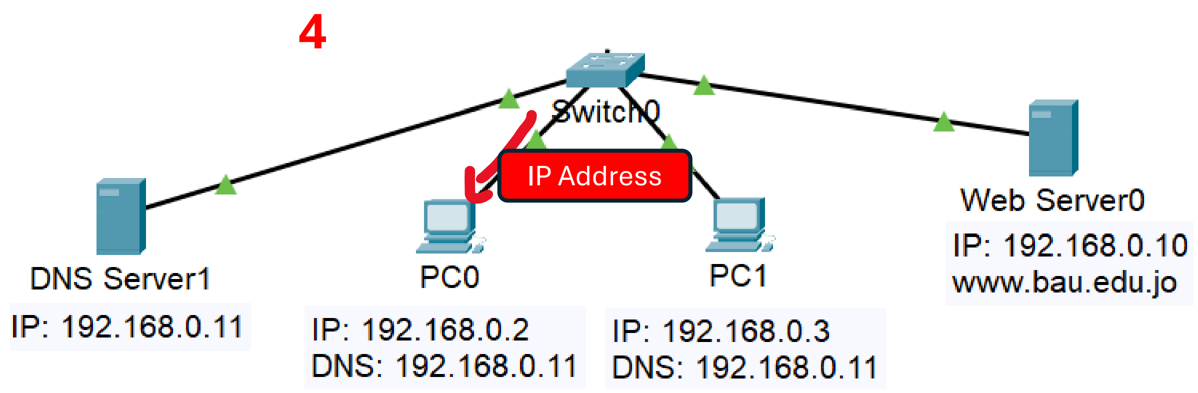



1
Switch0
Name
DNS Server1
IP: 192.168.0.11
PC0
IP: 192.168.0.2
DNS: 192.168.0.11
PC1
IP: 192.168.0.3
DNS: 192.168.0.11
Web Server0
IP: 192.168.0.10
www.bau.edu.jo

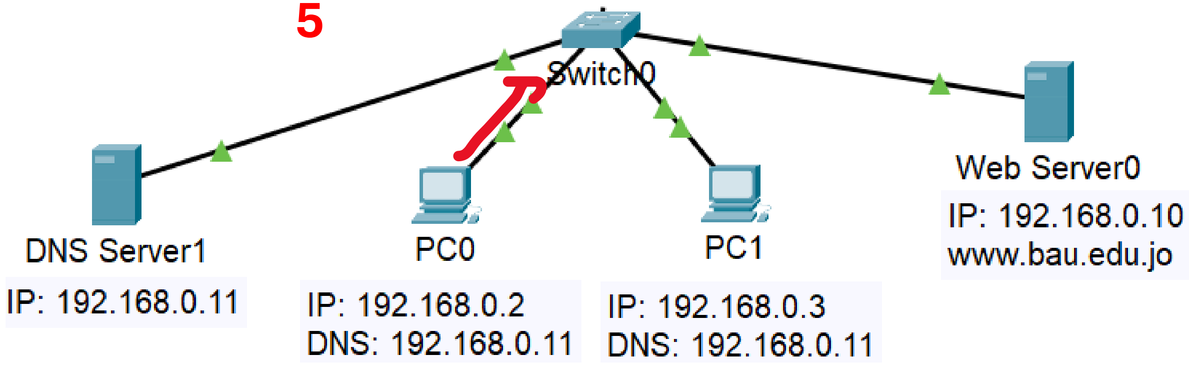

2
Name
Switch0
DNS Server1
IP: 192.168.0.11
Search in DNS Table

Name | IP Address
PC0
IP: 192.168.0.2
DNS: 192.168.0.11
PC1
IP: 192.168.0.3
DNS: 192.168.0.11
Web Server0
IP: 192.168.0.10
www.bau.edu.jo


3
IP
Address
Switch0
DNS Server1
IP: 192.168.0.11
PC0
IP: 192.168.0.2
DNS: 192.168.0.11
PC1
IP: 192.168.0.3
DNS: 192.168.0.11
Web Server0
IP: 192.168.0.10
www.bau.edu.jo

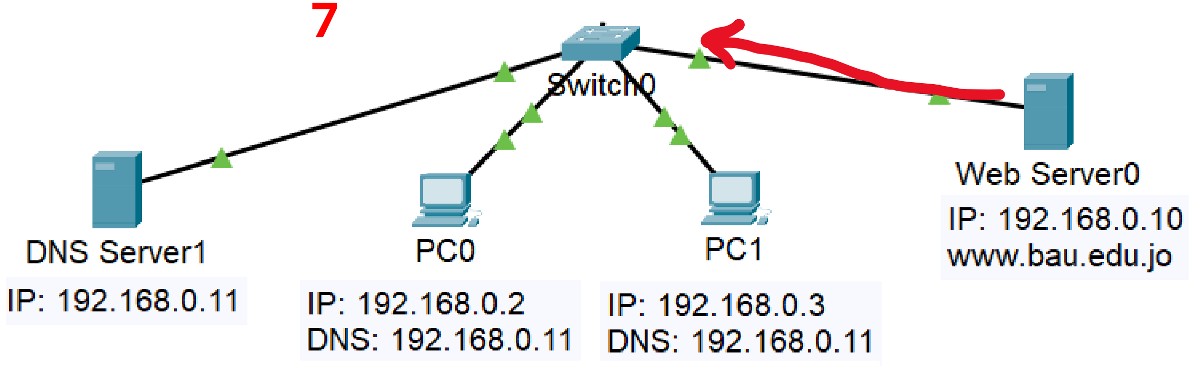

4
Switch0
IP Address
DNS Server1
IP: 192.168.0.11
PC0
IP: 192.168.0.2
DNS: 192.168.0.11
PC1
IP: 192.168.0.3
DNS: 192.168.0.11
Web Server0
IP: 192.168.0.10
www.bau.edu.jo

5



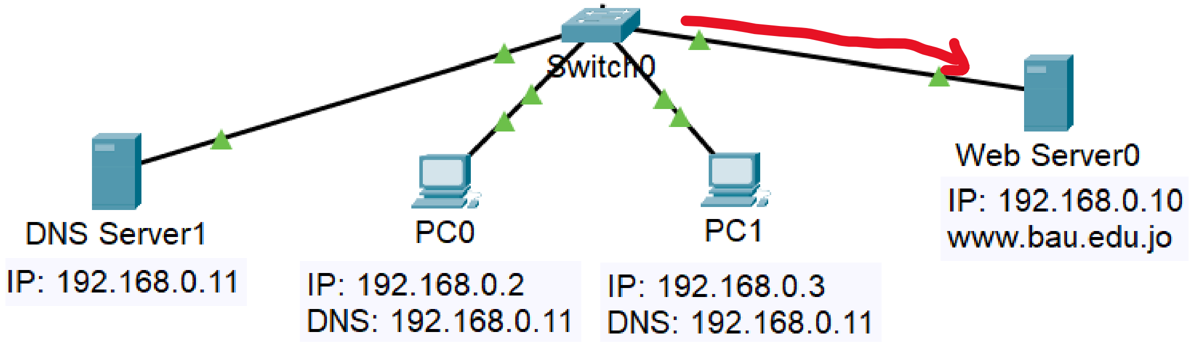
Switch0
DNS Server1
IP: 192.168.0.11
PC0
IP: 192.168.0.2
DNS: 192.168.0.11
PC1
IP: 192.168.0.3
DNS: 192.168.0.11
Web Server0
IP: 192.168.0.10
www.bau.edu.jo

7



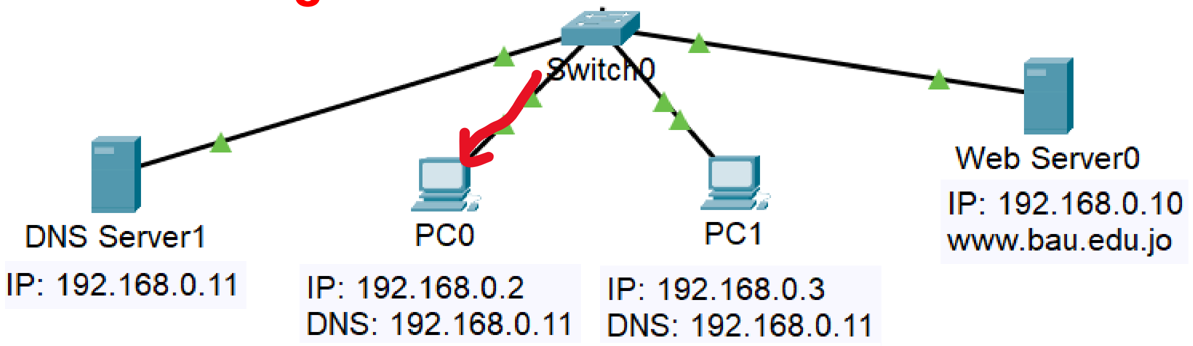
Switch0
DNS Server1
IP: 192.168.0.11
PC0
IP: 192.168.0.2
DNS: 192.168.0.11
PC1
IP: 192.168.0.3
DNS: 192.168.0.11
Web Server0
IP: 192.168.0.10
www.bau.edu.jo

6



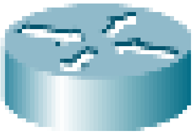
Switch0
DNS Server1
IP: 192.168.0.11
PC0
IP: 192.168.0.2
DNS: 192.168.0.11
PC1
IP: 192.168.0.3
DNS: 192.168.0.11
Web Server0
IP: 192.168.0.10
www.bau.edu.jo

8

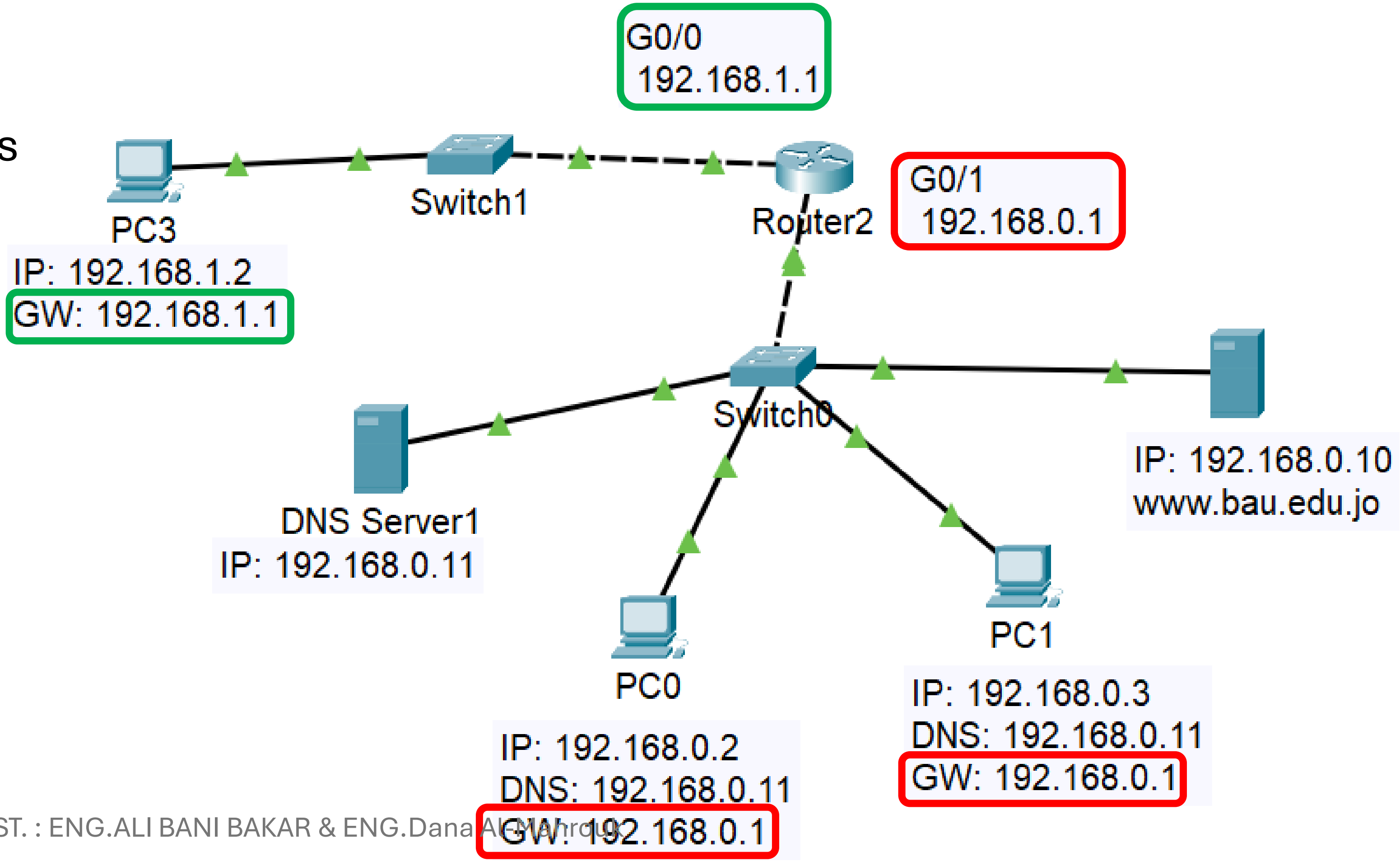


Switch0
DNS Server1
IP: 192.168.0.11
PC0
IP: 192.168.0.2
DNS: 192.168.0.11
PC1
IP: 192.168.0.3
DNS: 192.168.0.11
Web Server0
IP: 192.168.0.10
www.bau.edu.jo

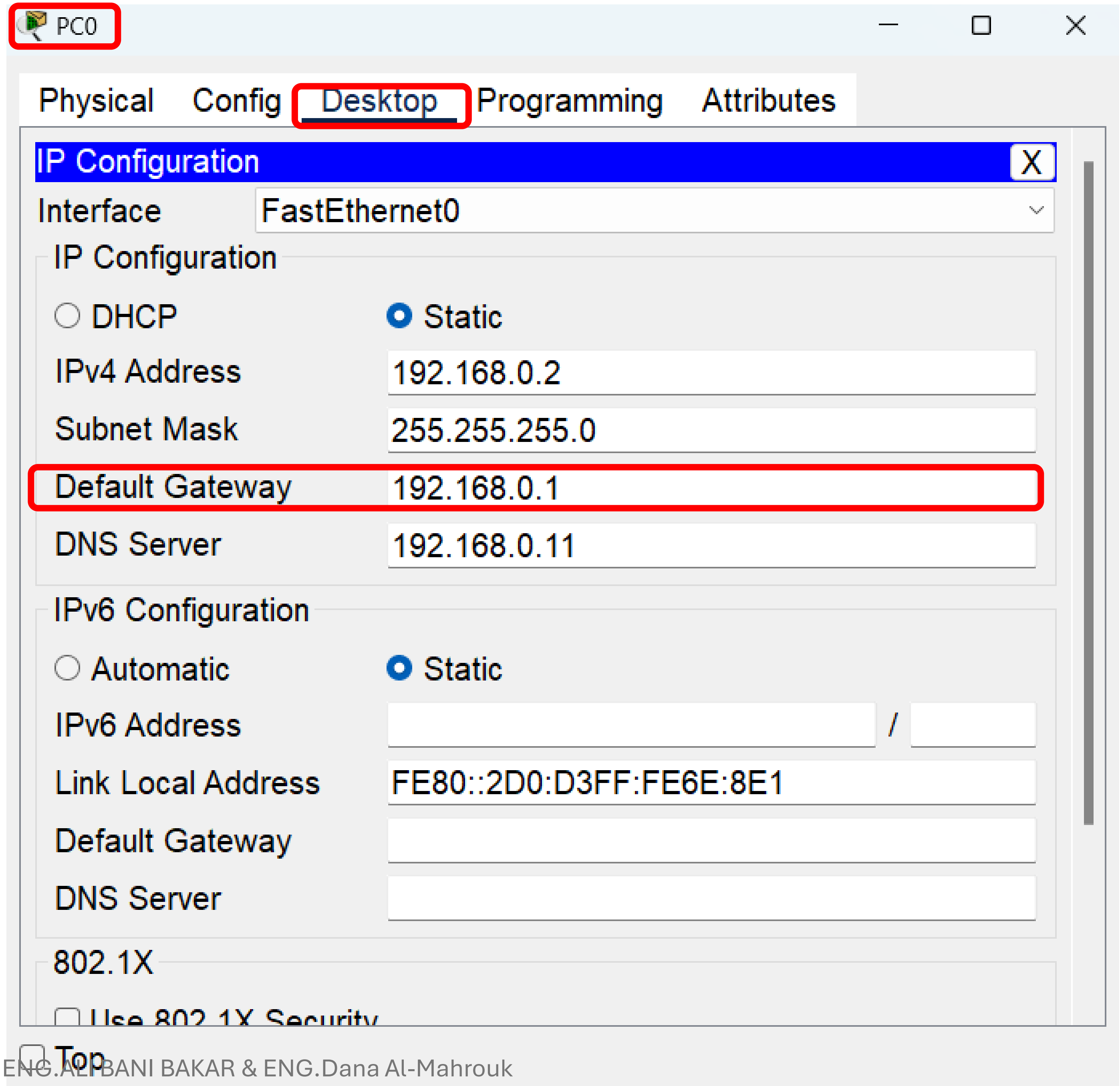
Router



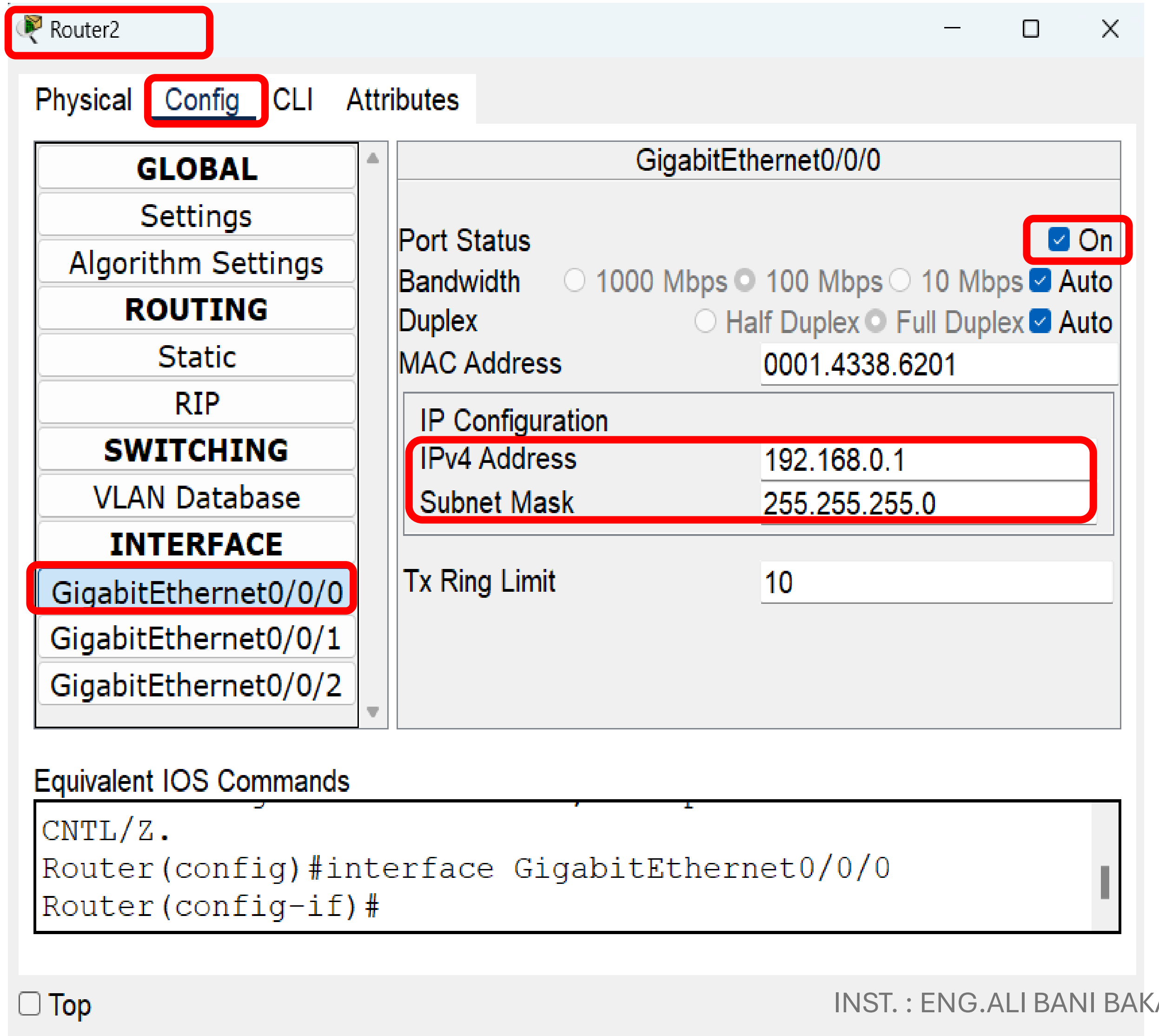
- Connects between two network.
- PC connects to other PC on different network using Router and put **Default Gateway** to IP Address of the **Router Interface IP** that connected to switch to that network.

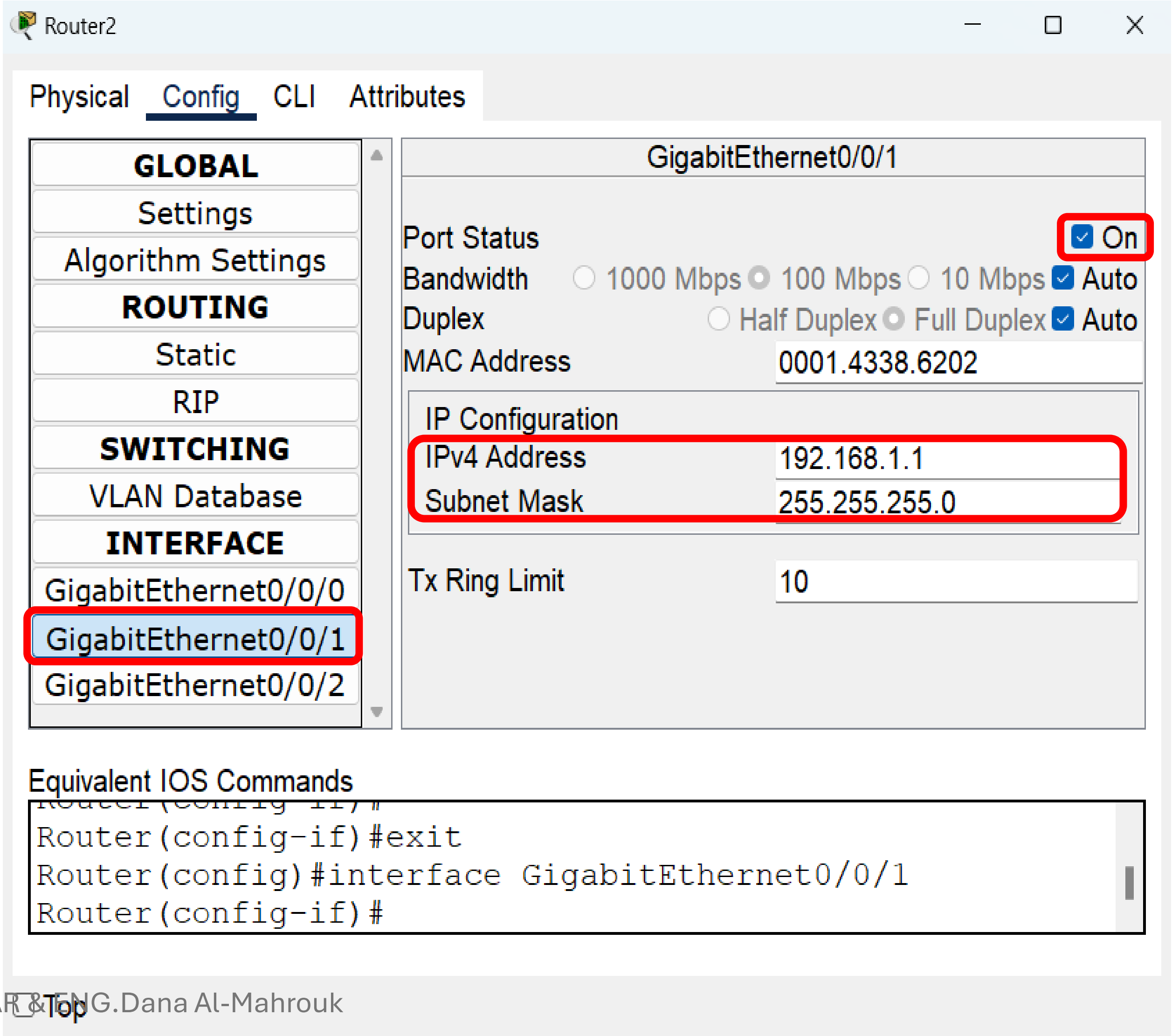


Give PC Default Gateway



Give Router IPs Address

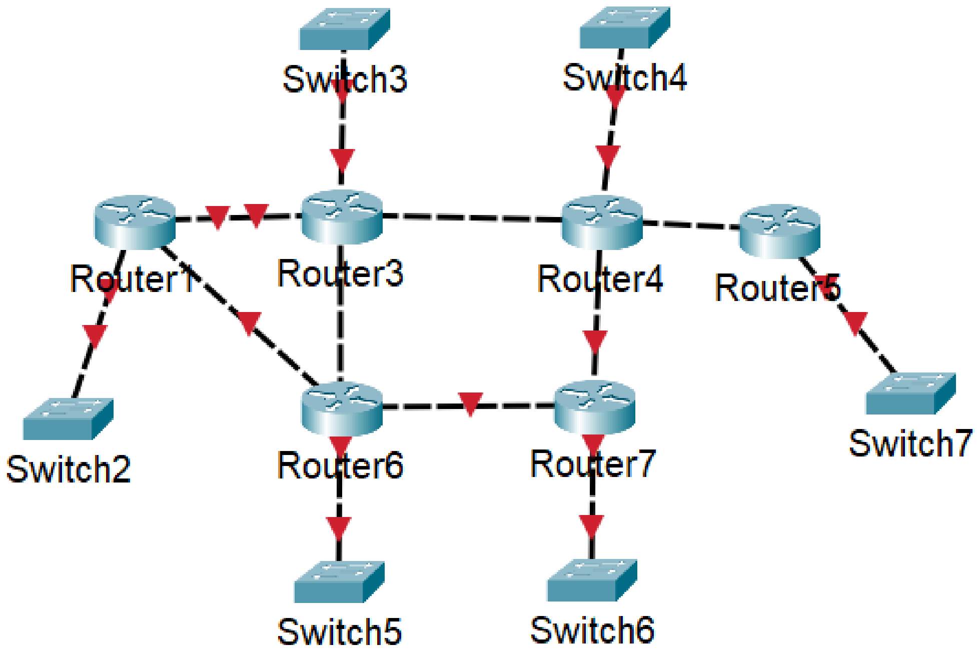


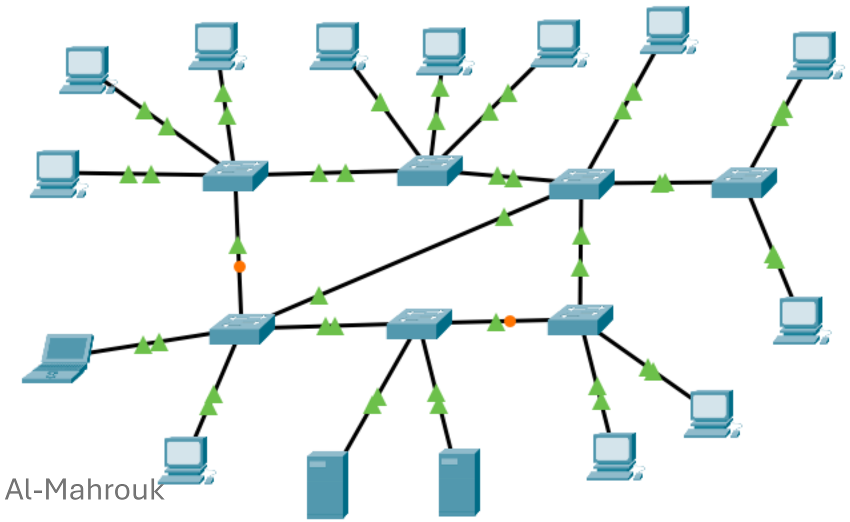


Question

- Q1: How many Networks in the typology?
- A: 13

- Q2: Can a PC or server can be configured to act as a default gateway in a network
- A: Yes, as firewall





- Q3 : How many Networks?
- Ans: 1

Day 2

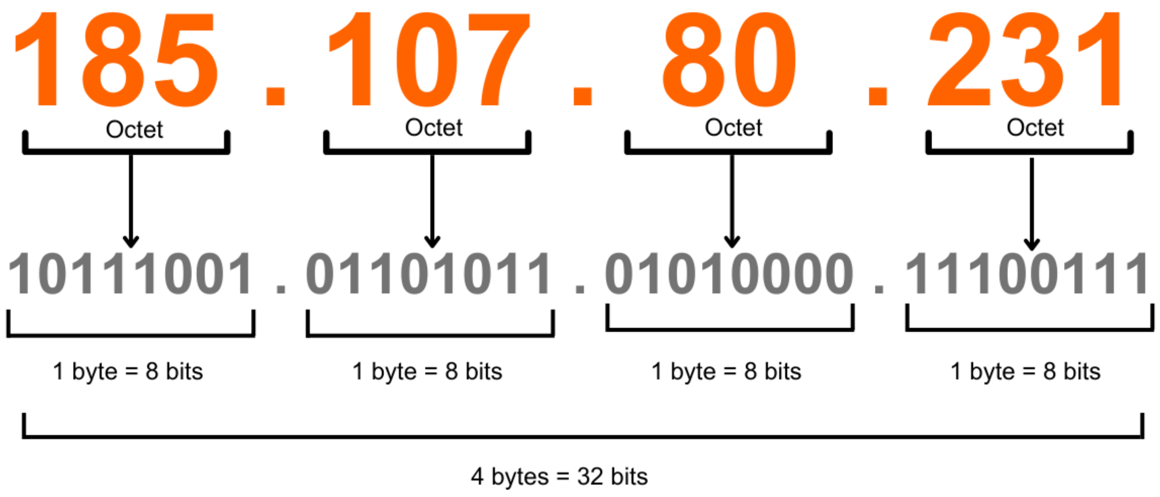
- Outline:
  - IPv4
  - Subnetting and CIDR
  - Network ID & Broadcast IP Examples
  - DHCP :APIPA & DHCP Server
  - How does DHCP work?
  - GNS3
  - Wireshark
  - Exercises

IPv4

- MAC Address → do not care about it, it changes during message sending – transform at each router.

- IPv4:

1. 4 digit → each 8-bit 1-byte Octet
2. 4 * 8-bit = 32-bit → 2^32 = 4 billions IP Address
3. 8-bit each → 2^8 = 256 → [0 - 255]
4. 0-255 . 0-255 . 0-255 . 0-255



- IPv6:
- 8 digit → each 16 bit | 4 hexadicemal → 128 bit
- ffff:ffff:ffff:ffff:ffff:ffff:ffff:fffff

- Q: why IPv4 still use? (the answer will be discussed later)

Network ID

- IP Address: 192.168.0.2 → 11000000.10101000.00000000.00000010
- Subnet Mask: 255.255.255.0 → 11111111.11111111.11111111.00000000

- **Network ID: [IP AND Subnet Mask]** → 192.168.0.0

Ex1:

- IP Address: 192.168.0.3 ➔ 11000000.10101000.00000000.00000011
- Subnet Mask: 255.255.255.0 ➔ 11111111.11111111.11111111.00000000

- **Network ID**: **[IP AND Subnet Mask]** ➔ 192.168.0.0
- **Broadcast IP** (Directed): **convert host part on IP address to 1's** ➔ last 8-bit ➔ 192.168.0.255

- **First Host: Network ID + 1** ➔ 192.168.0.1
- **Last Host: Broadcast IP – 1** ➔ 192.168.0.254

Ex2:

- IP Address: 192.168.3.5**/24** → 11000000.10101000.00000011.00000101
- Subnet Mask: 255.255.255.0 → 11111111.11111111.11111111.00000000

- Network ID: [IP **AND** Subnet Mask] → 192.168.3.0
- Broadcast IP (Directed): convert host part on IP address to 1's → last 8-bit → 192.168.3.255

- First Host: Network ID + 1 → 192.168.3.1
- Last Host: Broadcast IP – 1 → 192.168.3.254

Ex3:

- IP Address: 192.168.4.129**/25** → 11000000.10101000.00000010.10000001
- Subnet Mask: 255.255.255.128 → 11111111.11111111.11111111.10000000

- Network ID: [IP **AND** Subnet Mask] → 192.168.4.128
- Broadcast IP (Directed): convert host part on IP address to 1's → last 8-bit → 192.168.4.255

- First Host: Network ID + 1 → 192.168.4.129
- Last Host: Broadcast IP – 1 → 192.168.4.254

Ex4:

- IP Address: 192.168.4.64/**26** ➔ 11000000.10101000.00000010.01000000
- Subnet Mask: 255.255.255.192 ➔ 11111111.11111111.11111111.11000000

- Network ID: [IP **AND** Subnet Mask] ➔ 192.168.4.64
- Broadcast IP (Directed): convert host part on IP address to 1's ➔ last 8-bit ➔ 192.168.4.127

- First Host: Network ID + 1 ➔ 192.168.4.65
- Last Host: Broadcast IP – 1 ➔ 192.168.4.126

HW:

- IP Address: 192.168.5.33/**27** → 11000000.10101000.00000101.00100001
- Subnet Mask: 255.255.255.224 → 11111111.11111111.11111111.11100000

- Network ID: [IP **AND** Subnet Mask] → 192.168.5.32
- Broadcast IP (Directed): convert host part on IP address to 1's → last 8-bit →192.168.5.63

- First Host: Network ID + 1 → 192.168.5.31
- Last Host: Broadcast IP – 1 → 192.168.5.62

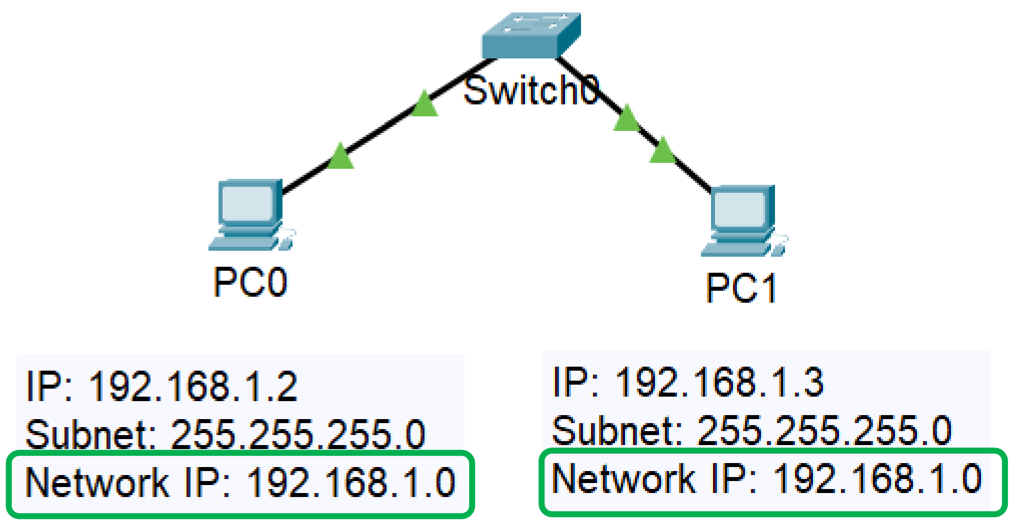
Ex5:

- IP Address: 192.168.10.2**/25** ➔ 11000000.10101000.00001010.10000010
- Subnet Mask: 255.255.255.128 ➔ 11111111.11111111.11111111.10000000

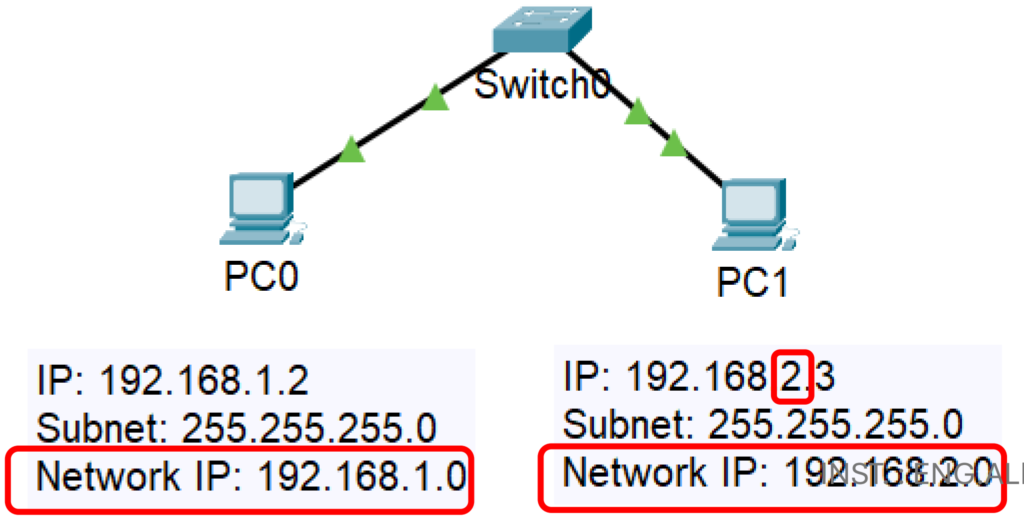
- Network ID: [IP **AND** Subnet Mask] ➔ 192.168.10.128
- Broadcast IP (Directed): convert host part on IP address to 1's ➔ last 8-bit ➔ 192.168.10.255

- First Host: Network ID + 1 ➔ 192.168.10.129
- Last Host: Broadcast IP – 1 ➔ 192.168.10.254

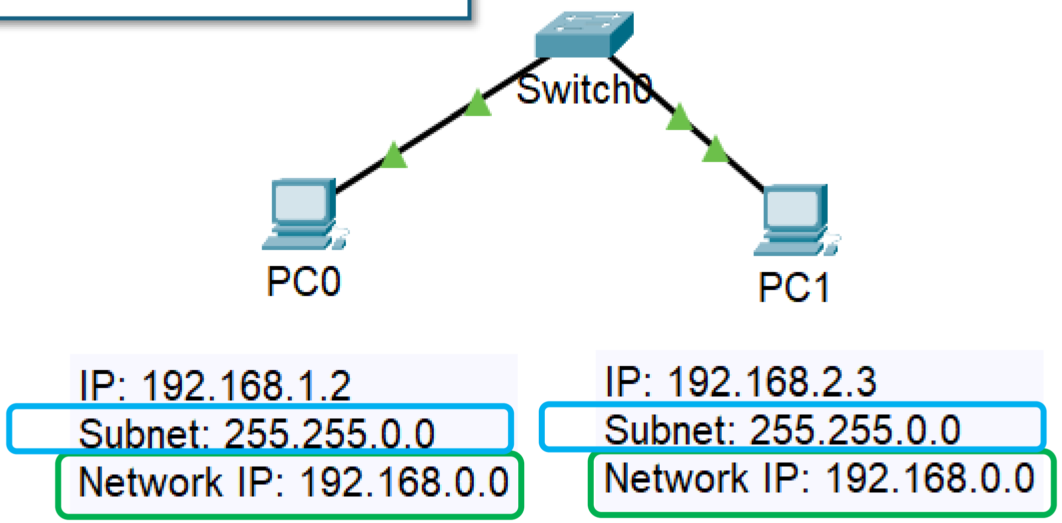
1) Connection Success



2) Connection fail



3) Connection Success



1) All PCs have the same Network, same Network ID ➔ 192.168.1.0
2) All PCs have not the same Network, have not same Network ID

- Network ID 1 ➔ 192.168.1.0
- Network ID 2 ➔ 192.168.2.0
- [PC can't connection]

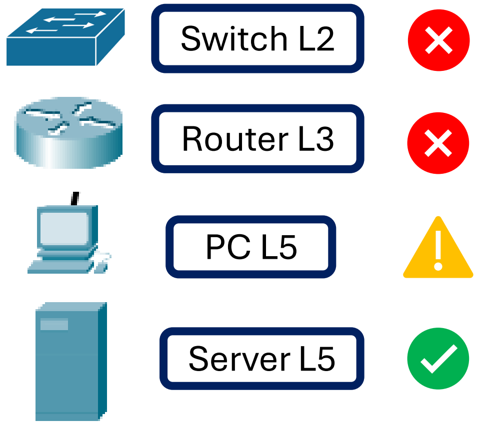
3) All PCs have the same Network, same Network ID ➔ 192.168.0.0

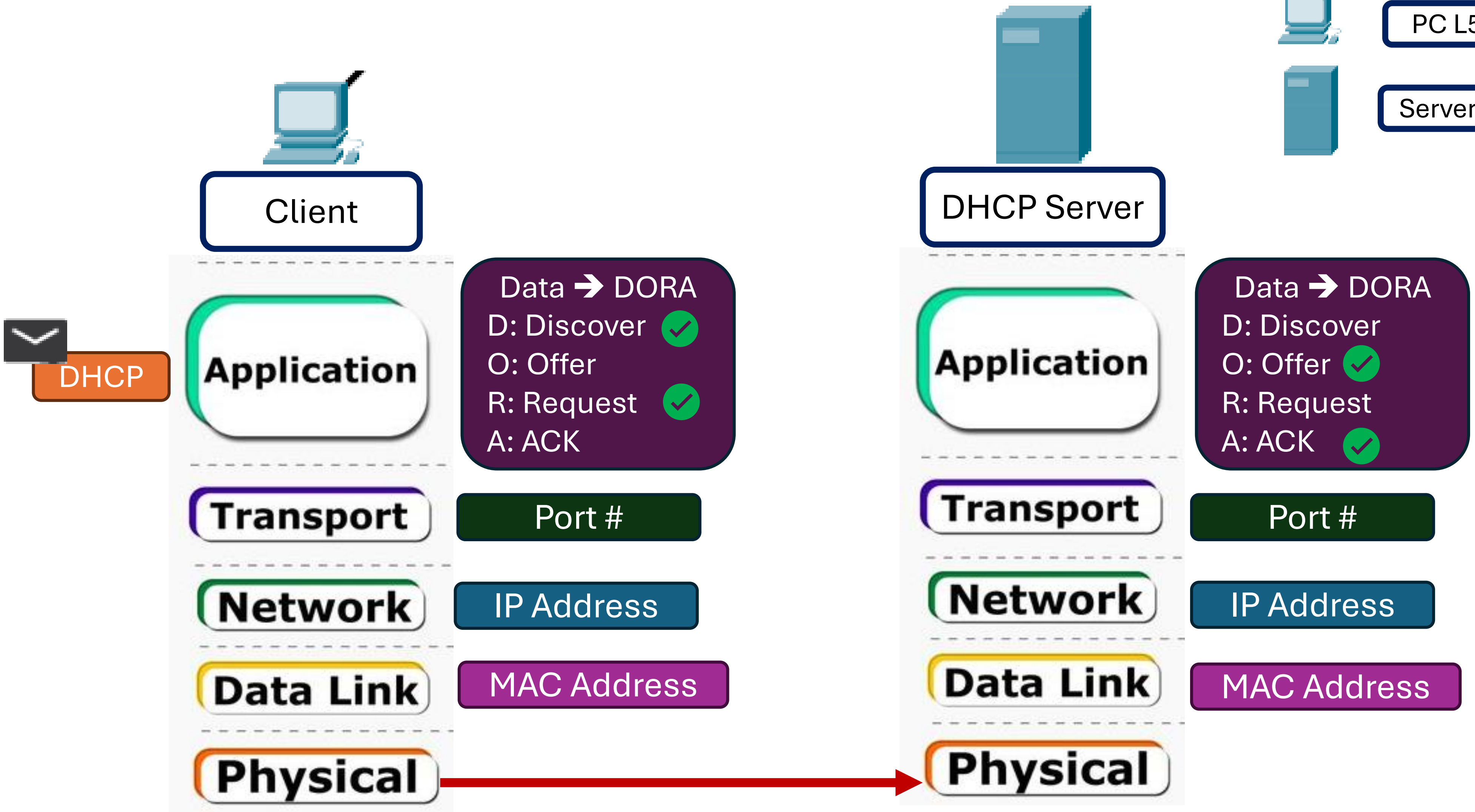
Change subnet mask for all PC to ➔ 255.255.0.0

DHCP Protocol

- **Dynamic** Host Configuration Protocol (**DHCP**): **Application Layer 5**
- provides services that support **network applications**. It handles the assignment of **IP addresses and network configuration to devices**, enabling them to communicate on the network.
- Give Host (PC) network configuration :
  - IP Address
  - Default Gateway
  - DNS Server
  - Subnet Mask
- Automatic Private IP Addressing (**APIPA)**: when there no DHCP Server, PC give itself a network configuration when we choose DHCP.

DHCP Device?


Switch L2
Router L3
PC L5
Server L5


Client
DHCP
Application
Data → DORA
D: Discover
O: Offer
R: Request
A: ACK
Transport
Port #
Network
IP Address
Data Link
MAC Address
Physical
DHCP Server
Application
Data → DORA
D: Discover
O: Offer
R: Request
A: ACK
Transport
Port #
Network
IP Address
Data Link
MAC Address
Physical

DHCP Server

- It gives you range of hosts.
- It Does not give itself an IP like other PC, you must give it a **static IP**.
- You must not allow to give a **Server IP** from DHCP server ➔ server must have a constant IP ➔ **static IP** – manually.
- You can put a list of **exclusive** IPs Address; that DHCP will not give it to any hosts.

- IP address of hosts is dynamic ➔ it change in some cases:
  - End connection with a network
  - Shutdown the host
  - Lease time ended

How it work?

- Discover DHCP server (port# 67) ➔ **DORA**, to get dynamic host configuration network to PC.
- DHCP server is a give a **Network Services**, at main port at server side not client side.

1. **Discover**: The **client** sends out a DHCP Discover message to **find available DHCP servers**. This is a **broadcast** message in local network ➔

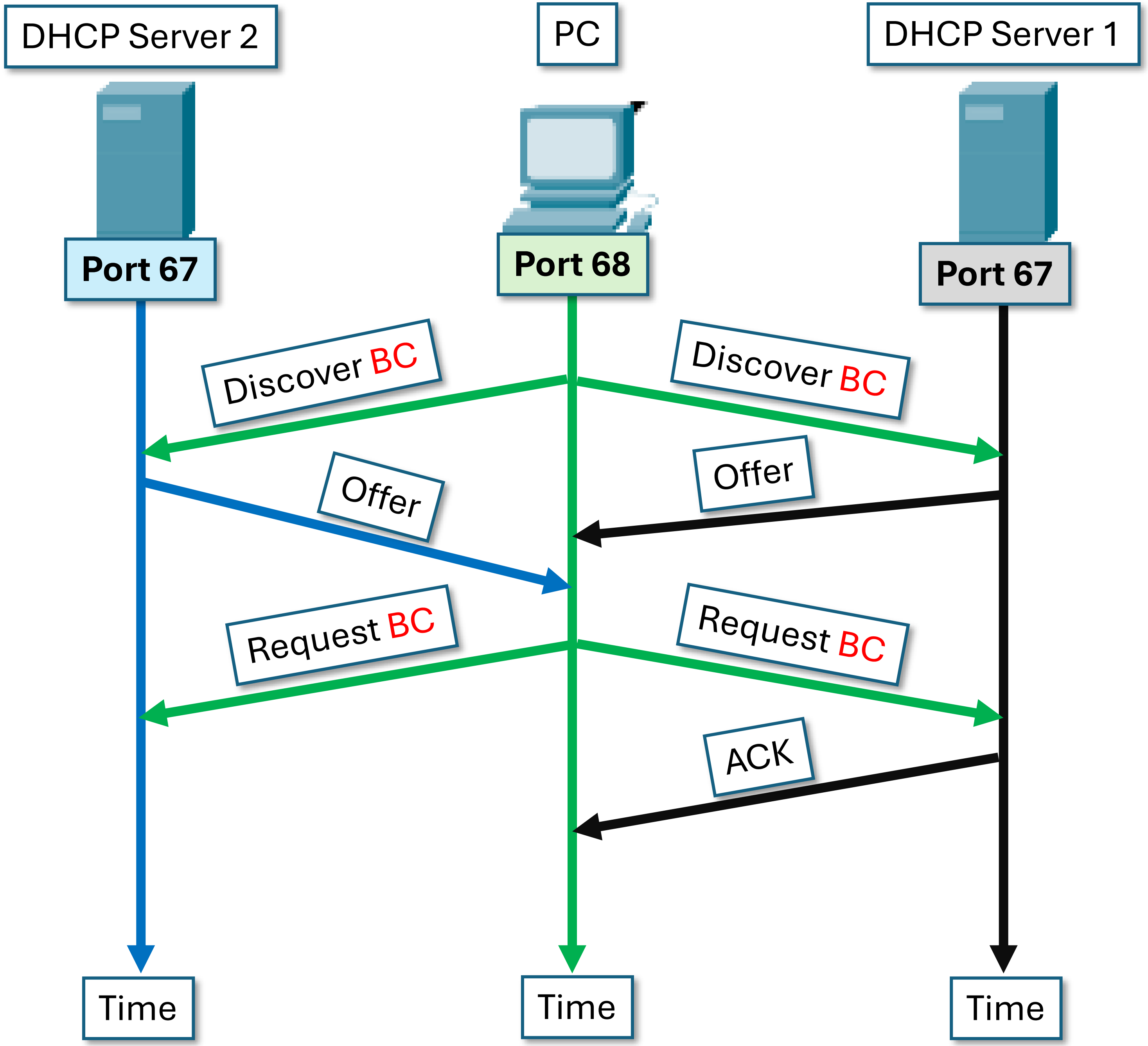
IPv4 is Src: 0.0.0.0 & Dest: 255.255.255.255.
MAC is Src: it own MAC & Dest: FF:FF:FF:FF:FF:FF

- Q: what is DHCP Relay?

2. **Offer**: **One or more DHCP servers respond** with a DHCP Offer message. This message contains an available IP address and other network configuration details.

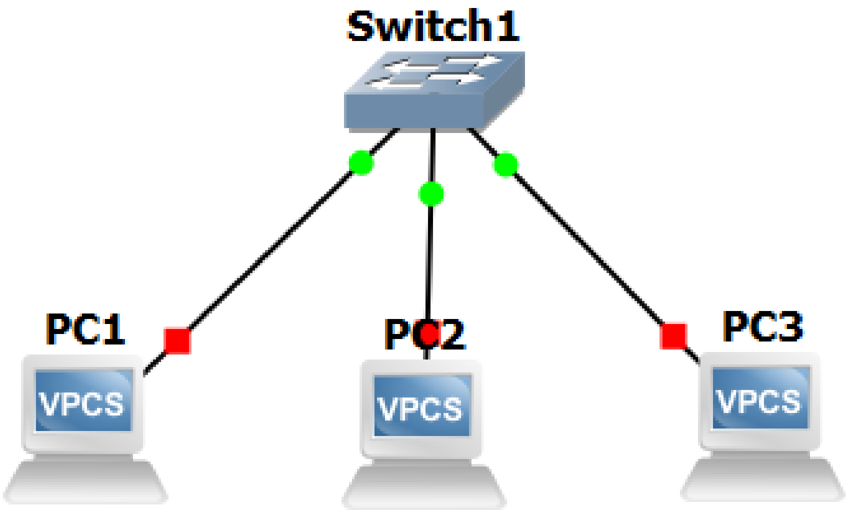
3. **Request**: The **client responds** to the server with a DHCP Request message, indicating that it wants to **accept** the offered IP address and network configuration.

4. **Acknowledgement** : The DHCP server sends a DHCP Acknowledge message to **confirm** that the IP address and network configuration have been assigned to the client. At this point, the client can use the assigned IP address to communicate on the network.


DHCP Server 2
PC
DHCP Server 1
Port 67
Port 68
Port 67
Discover BC
Discover BC
Offer
Offer
Request BC
Request BC
ACK
Time
Time
Time

GNS3

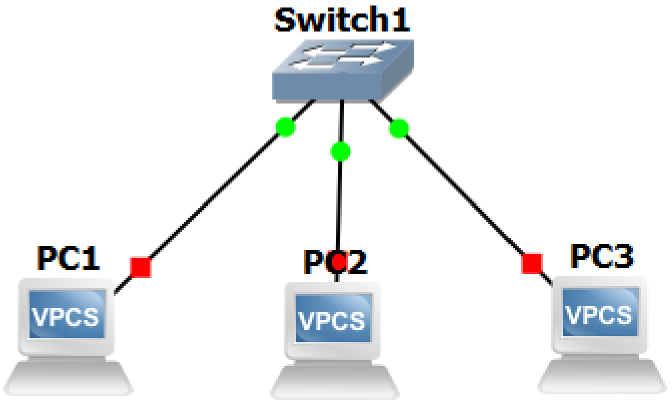
GNS3 (Graphical Network Simulator-3)
is a network software emulator that allows users to simulate, configure, test, and troubleshoot complex networks using virtual devices.

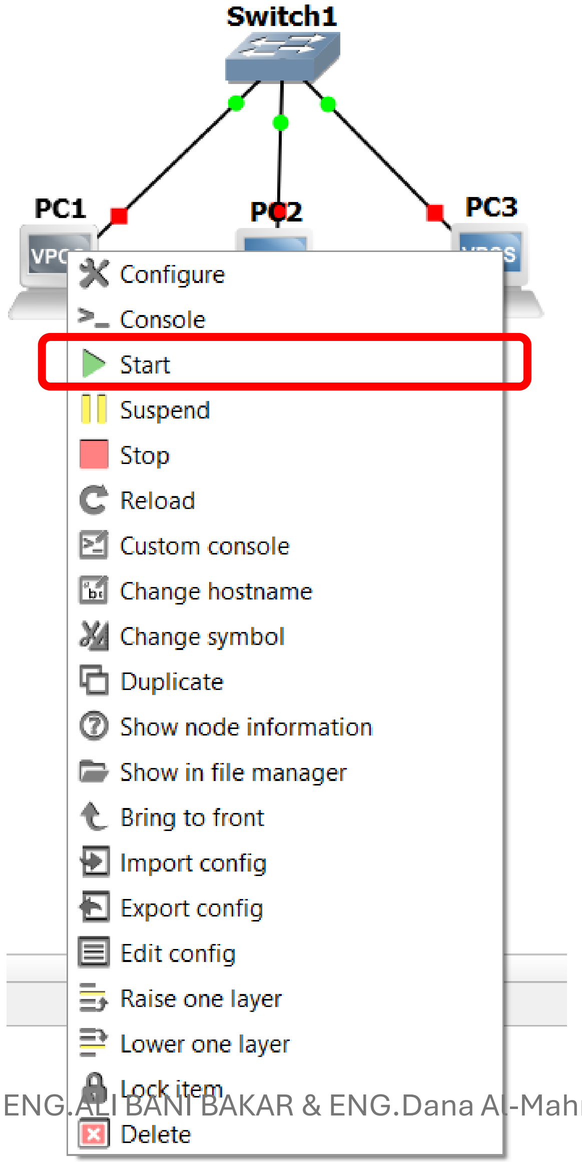


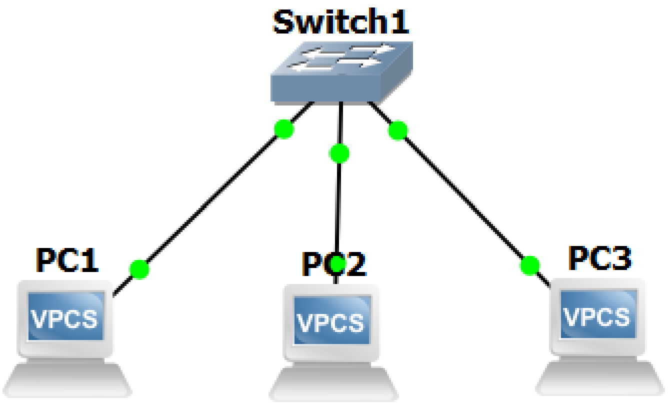
Wireshark

Wireshark is a popular **open-source network protocol analyzer** used for **capturing and analyzing network traffic** in real-time. It helps network administrators and security professionals **troubleshoot** network issues, **monitor** network performance, and **detect** security **vulnerabilities**.

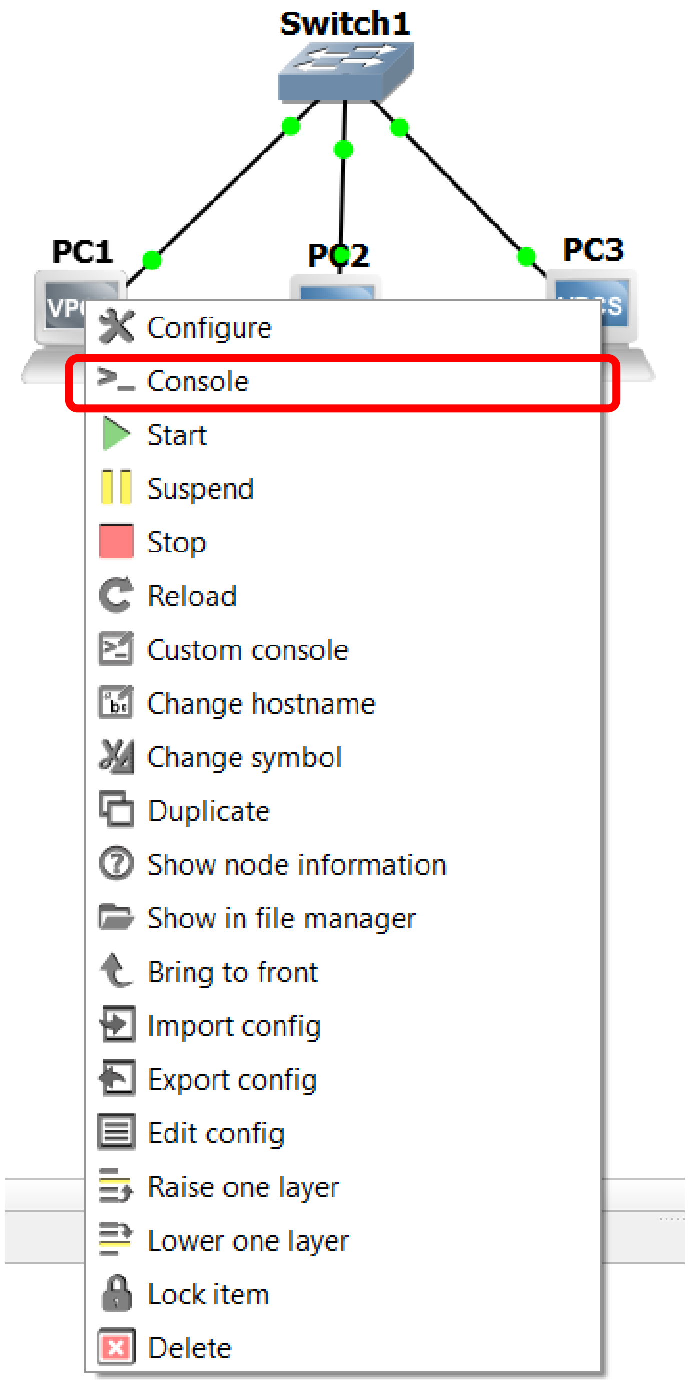
ARP example using GNS3







By default, PC is turn Off
You must turn it On



Open Console in PC

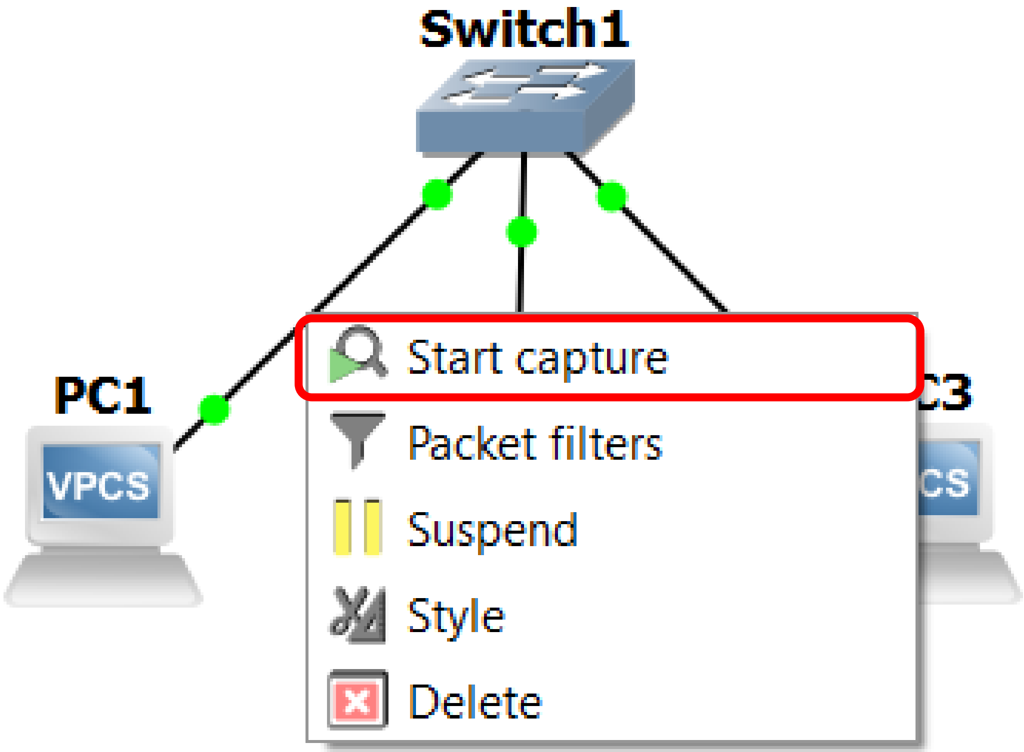
```
PC1> ip 192.168.0.2 255.255.255.0
Checking for duplicate address...
PC1 : 192.168.0.2 255.255.255.0
```

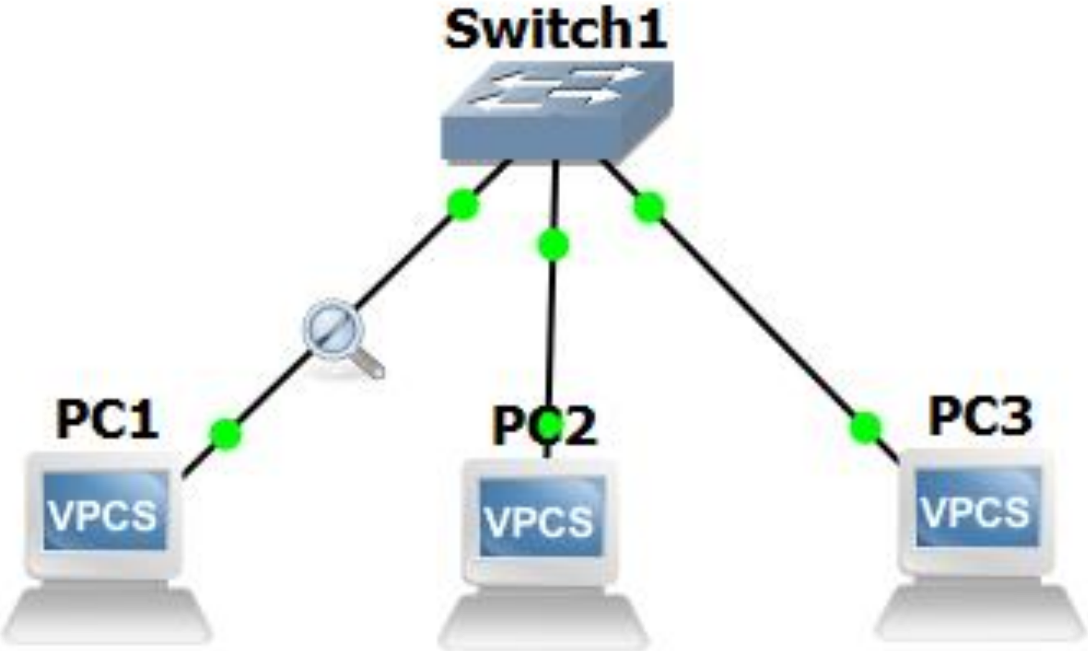
```
PC2> ip 192.168.0.3 255.255.255.0
Checking for duplicate address...
PC1 : 192.168.0.3 255.255.255.0
```

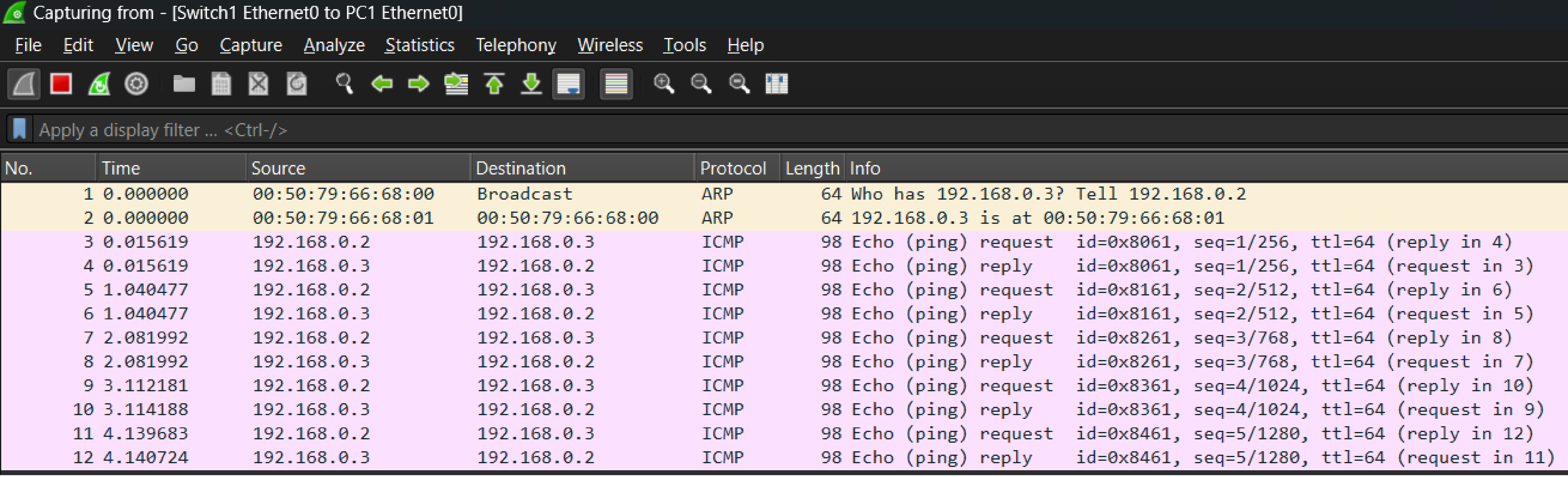
```
PC3> ip 192.168.0.4 255.255.255.0
Checking for duplicate address...
PC1 : 192.168.0.4 255.255.255.0
```

```
PC1> ping 192.168.0.3
84 bytes from 192.168.0.3 icmp_seq=1 ttl=64 time=1.670 ms
84 bytes from 192.168.0.3 icmp_seq=2 ttl=64 time=2.817 ms
84 bytes from 192.168.0.3 icmp_seq=3 ttl=64 time=2.840 ms
84 bytes from 192.168.0.3 icmp_seq=4 ttl=64 time=2.530 ms
84 bytes from 192.168.0.3 icmp_seq=5 ttl=64 time=3.043 ms
```

Start Capture PC wire in Wireshark







No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:50:79:66:68:00	Broadcast	ARP	64	Who has 192.168.0.3? Tell 192.168.0.2
2	0.000000	00:50:79:66:68:01	00:50:79:66:68:00	ARP	64	192.168.0.3 is at 00:50:79:66:68:01
3	0.015619	192.168.0.2	192.168.0.3	ICMP	98	Echo (ping) request id=0x8061, seq=1/256, ttl=64 (reply in 4)
4	0.015619	192.168.0.3	192.168.0.2	ICMP	98	Echo (ping) reply id=0x8061, seq=1/256, ttl=64 (request in 3)
5	1.040477	192.168.0.2	192.168.0.3	ICMP	98	Echo (ping) request id=0x8161, seq=2/512, ttl=64 (reply in 6)
6	1.040477	192.168.0.3	192.168.0.2	ICMP	98	Echo (ping) reply id=0x8161, seq=2/512, ttl=64 (request in 5)
7	2.081992	192.168.0.2	192.168.0.3	ICMP	98	Echo (ping) request id=0x8261, seq=3/768, ttl=64 (reply in 8)
8	2.081992	192.168.0.3	192.168.0.2	ICMP	98	Echo (ping) reply id=0x8261, seq=3/768, ttl=64 (request in 7)
9	3.112181	192.168.0.2	192.168.0.3	ICMP	98	Echo (ping) request id=0x8361, seq=4/1024, ttl=64 (reply in 10)
10	3.114188	192.168.0.3	192.168.0.2	ICMP	98	Echo (ping) reply id=0x8361, seq=4/1024, ttl=64 (request in 9)
11	4.139683	192.168.0.2	192.168.0.3	ICMP	98	Echo (ping) request id=0x8461, seq=5/1280, ttl=64 (reply in 12)
12	4.140724	192.168.0.3	192.168.0.2	ICMP	98	Echo (ping) reply id=0x8461, seq=5/1280, ttl=64 (request in 11)

- Here we see the **capture analysis** that we placed on the first computer wire.
- First, the device searches for the recipient's MAC address in the **ARP table**, because for the first time the table is **empty**, and it will not find the address.
- The first message is the **ARP message**. The device sends a **broadcast** to all devices asking who owns this IP address, so the receiver sends the MAC address to the sender IP address.
- The response is **unicast** to the sender and contains the recipient **MAC address** of the message.

- The computer saves this address at ARP Table and **begins sending the actual message**.

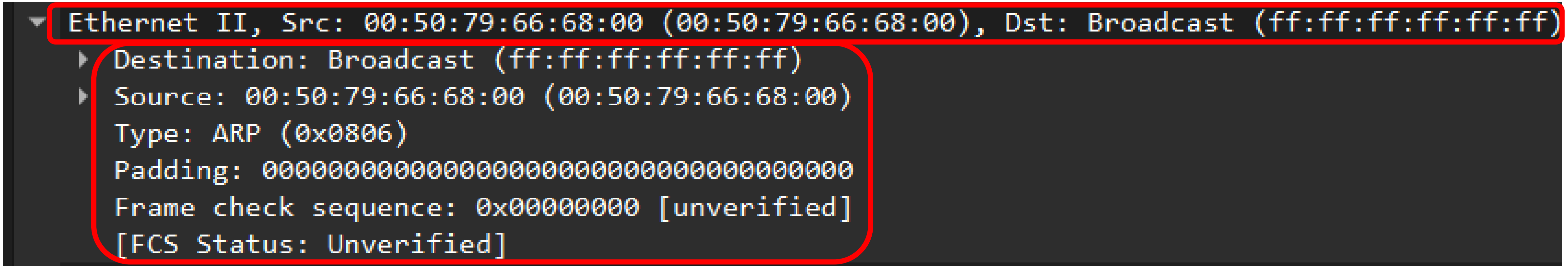
Wireshark · Packet 1 · -

```
Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
Ethernet II, Src: 00:50:79:66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
```

```
Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
    Section number: 1
    Interface id: 0 (-)
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul 16, 2024 15:03:12.253613000 Jordan Standard Time
    UTC Arrival Time: Jul 16, 2024 12:03:12.253613000 UTC
    Epoch Arrival Time: 1721131392.253613000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 64 bytes (512 bits)
    Capture Length: 64 bytes (512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
```

Here is the contents of the first frame sent, the Broadcast frame.

Frame → Physical L1
Ethernet II → Data Link L2
Address Resolution Protocol → message protocol



- **Ethernet II**: The frame uses the Ethernet II encapsulation.
- **Source**: `00:50:79:66:68:00` - The **MAC address** of the source device sending the frame.
- **Destination**: `ff:ff:ff:ff:ff:ff` - This indicates a **broadcast address**, meaning the frame is sent to all devices on the local network.
- **Type**: `ARP (0x0806)` - The frame contains an ARP (Address Resolution Protocol) packet.
- **Padding**: `000000000000000000000000000000000000` - Additional data added to meet the **minimum frame size** requirement.
- **Frame check sequence (FCS)**: `0x00000000 [unverified]` - The FCS is used to check the integrity of the frame. Here, it shows as unverified, meaning the checksum has not been validated.
- **FCS Status**: Unverified - Indicates the frame check sequence status has not been verified.

```
▾ Address Resolution Protocol (request)
      Hardware type: Ethernet (1)
      Protocol type: IPv4 (0x0800)
      Hardware size: 6
      Protocol size: 4
      Opcode: request (1)
      Sender MAC address: 00:50:79:66:68:00 (00:50:79:66:68:00)
      Sender IP address: 192.168.0.2
      Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
      Target IP address: 192.168.0.3
```

Wireshark · Packet 3 · -

```
Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
Ethernet II, Src: 00:50:79:66:68:00 (00:50:79:66:68:00), Dst: 00:50:79:66:68:01 (00:50:79:66:68:01)
Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.3
Internet Control Message Protocol
```

```
Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
    Section number: 1
    Interface id: 0 (-)
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul 16, 2024 15:03:12.269232000 Jordan Standard Time
    UTC Arrival Time: Jul 16, 2024 12:03:12.269232000 UTC
    Epoch Arrival Time: 1721131392.269232000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.015619000 seconds]
    [Time delta from previous displayed frame: 0.015619000 seconds]
    [Time since reference or first frame: 0.015619000 seconds]
    Frame Number: 3
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
```

```
▾ Ethernet II, Src: 00:50:79:66:68:00 (00:50:79:66:68:00), Dst: 00:50:79:66:68:01 (00:50:79:66:68:01)
  ▸ Destination: 00:50:79:66:68:01 (00:50:79:66:68:01)
  ▸ Source: 00:50:79:66:68:00 (00:50:79:66:68:00)
    Type: IPv4 (0x0800)
```

```
▾ Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.3
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x6180 (24960)
  ▸ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x97d3 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.0.2
    Destination Address: 192.168.0.3
```

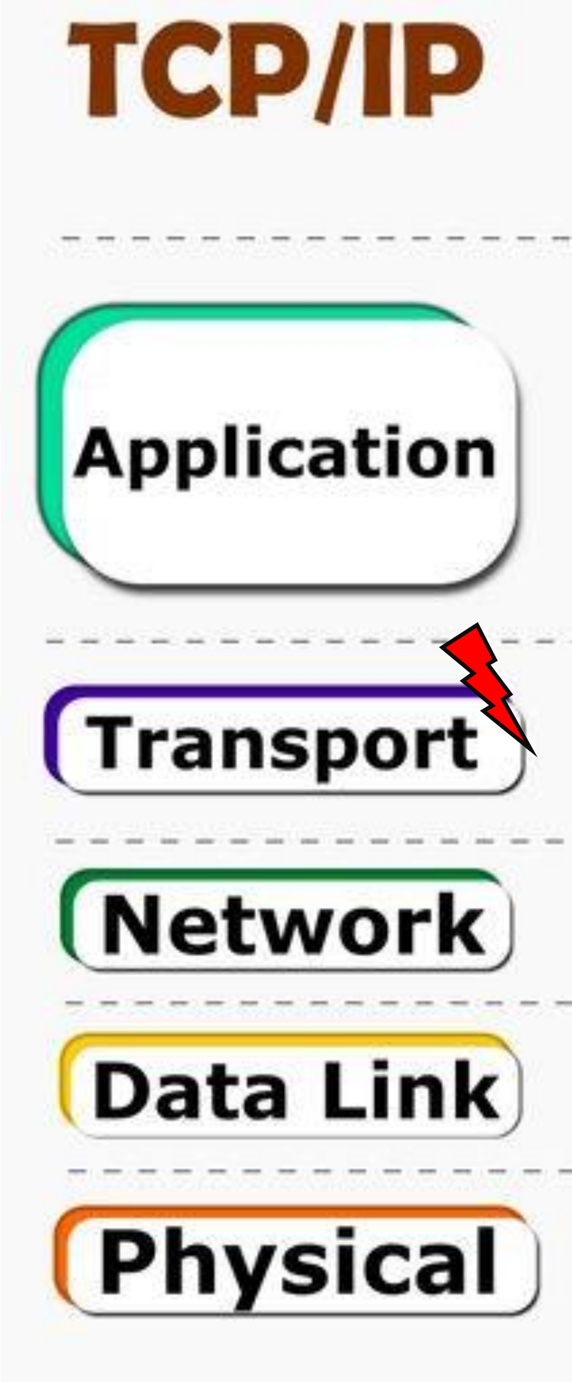
```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x9fa9 [correct]
    [Checksum Status: Good]
    Identifier (BE): 32865 (0x8061)
    Identifier (LE): 24960 (0x6180)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
    [Response frame: 4]
  Data (56 bytes)
```

Day 3

- Outline:
  - Transport Layer
  - Multiplexing
  - Subnetting

Transport Layer

- providing **end-to-end communication** services for applications. It ensures complete data transfer and manages error correction, data flow, and data segmentation.
- Protocols:
  - TCP
  - UDP
- Key Functions:
  - **Multiplexing and Demultiplexing ➔**
    - **Allows multiple applications to use the network simultaneously.**
    - **Uses port numbers to direct incoming data**
  - **Segmentation** and Reassembly ➔ Data divided & reassembled(TCP)
  - Connection Establishment and Termination ➔ connection/less & handshake
  - **Flow Control** ➔ Manages the rate of data transmission (TCP window-based)
  - **Error Detection** and Correction ➔ data integrity (TCP ACK & retransmissions)



Port

- **Client**: does not provide a service, he uses a **random port** that is not reserved for a service.
- **Server**: provides services, uses a **port designated for this service and continues listening on it** (waiting for the client to request the service from this port)

- There are three **ranges:** ➔ $[(2^{0}-1) - (2^{16}-1)]$
  - Well-Known Ports: 0 – 1023
  - Registered Ports: 1024 – 49151
  - Dynamic and Private Ports: 49152 – 65535 (Client used).

- Note:

The port number is **determined** in the application layer, and is **placed in the control data** in the transport layer

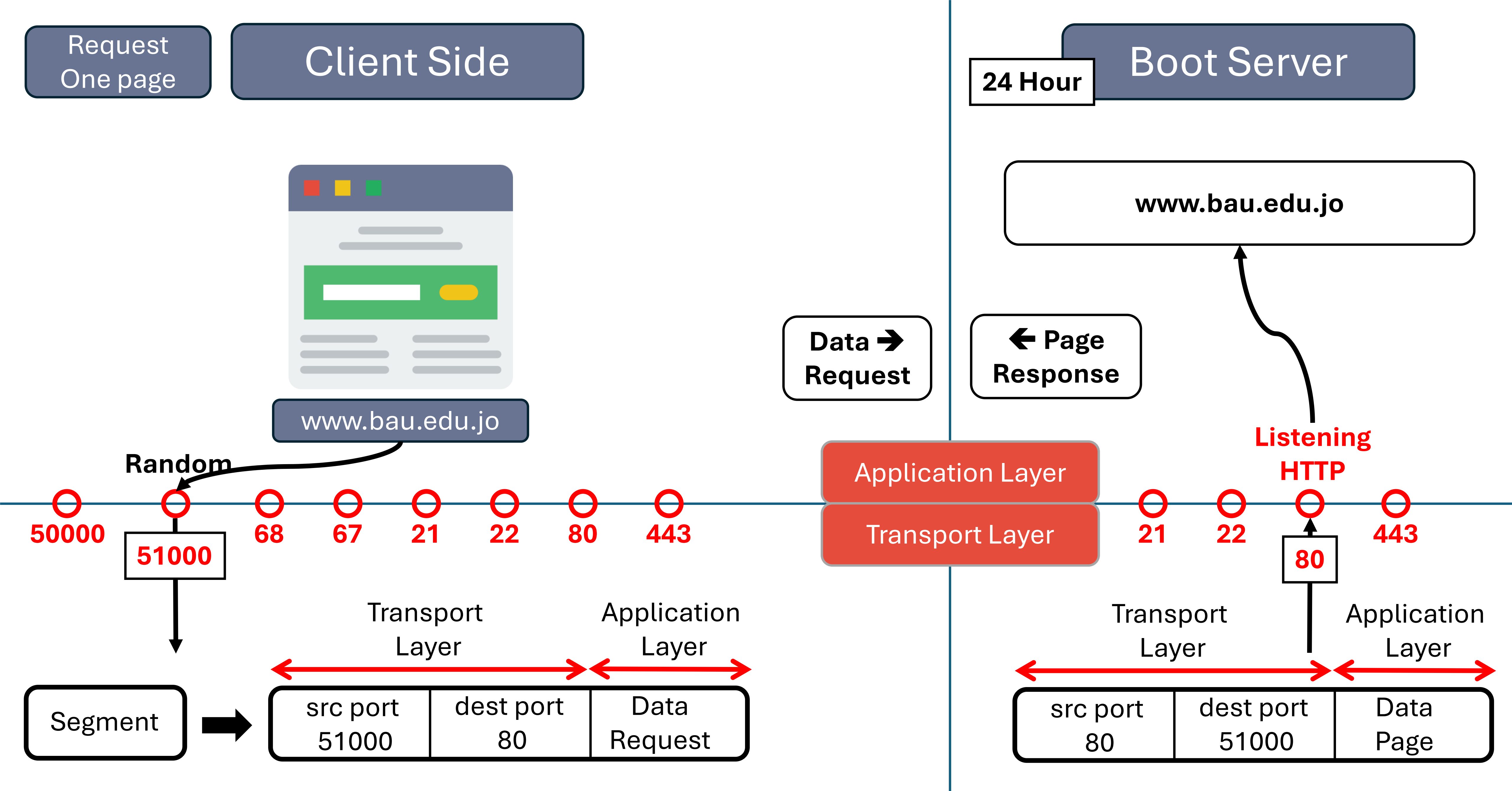
WELL-KNOWN COMMON PROTOCOLS	
20 & 21	FTP
80	HTTP
443	HTTPS
389	LDAP
636	LDAP (SSL)
161	SNMP
22	SSH
23	Telnet
25	SMTP
3389	Microsoft RDP
53	DNS Service
119	NNTP
143	IMAP
993	IMAP (SSL)
53	DNS
67	DHCP server
68	DHCP CLIENT

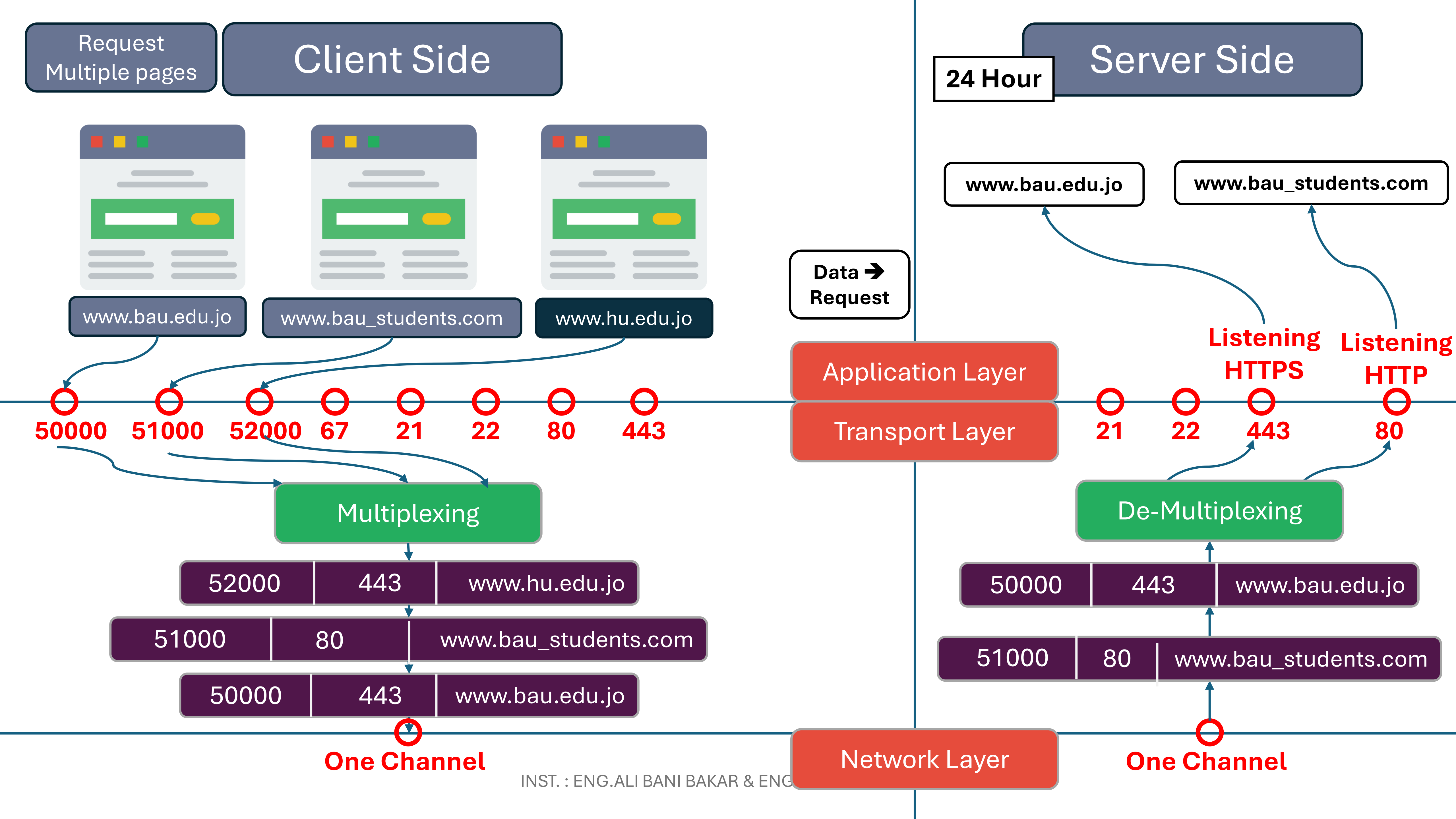
Question

- Q: Why is it ineffective to scan all ports by a hacker on a client?
- A:

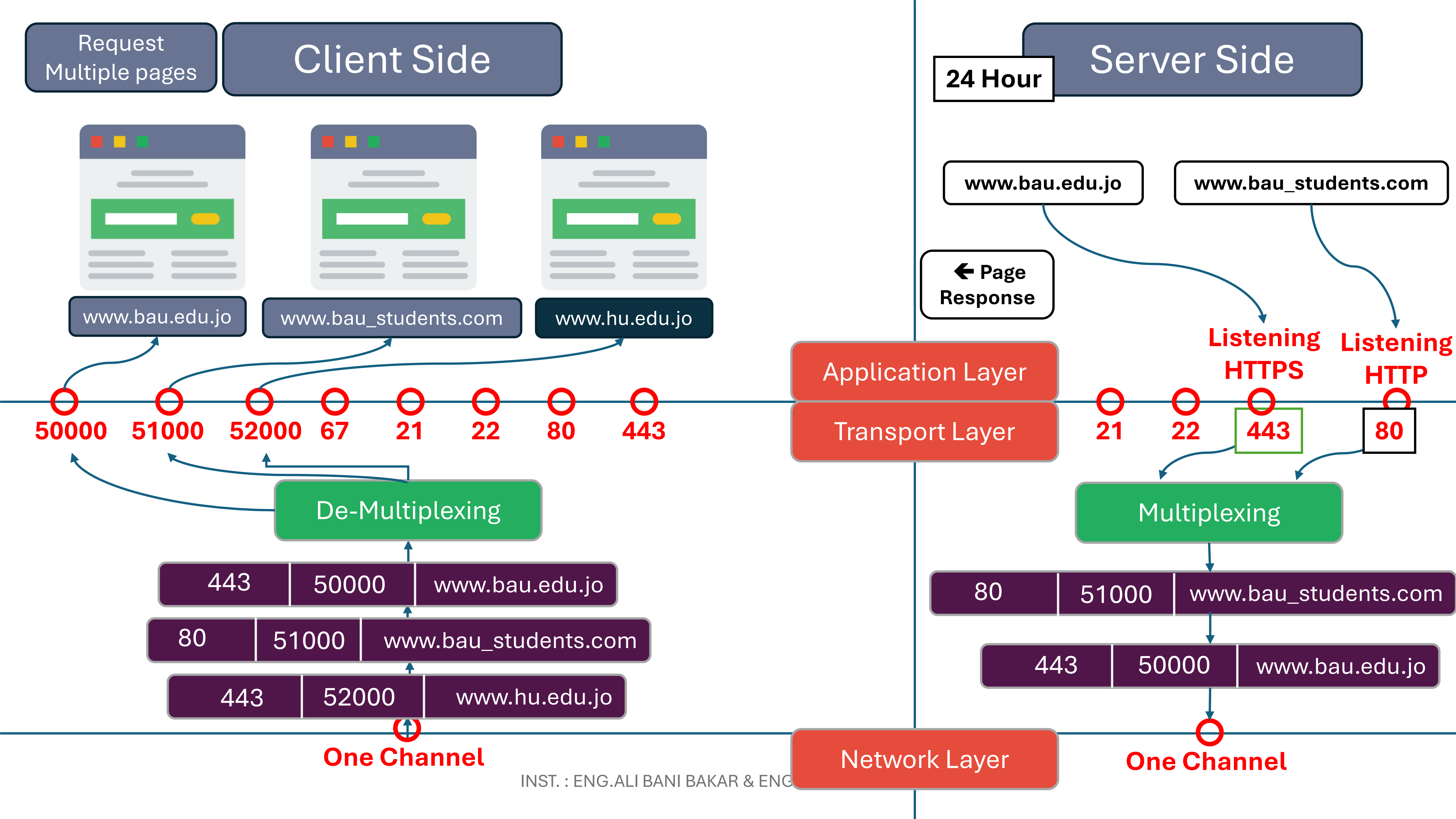
1) There are many ports and checking them all takes a lot of time.

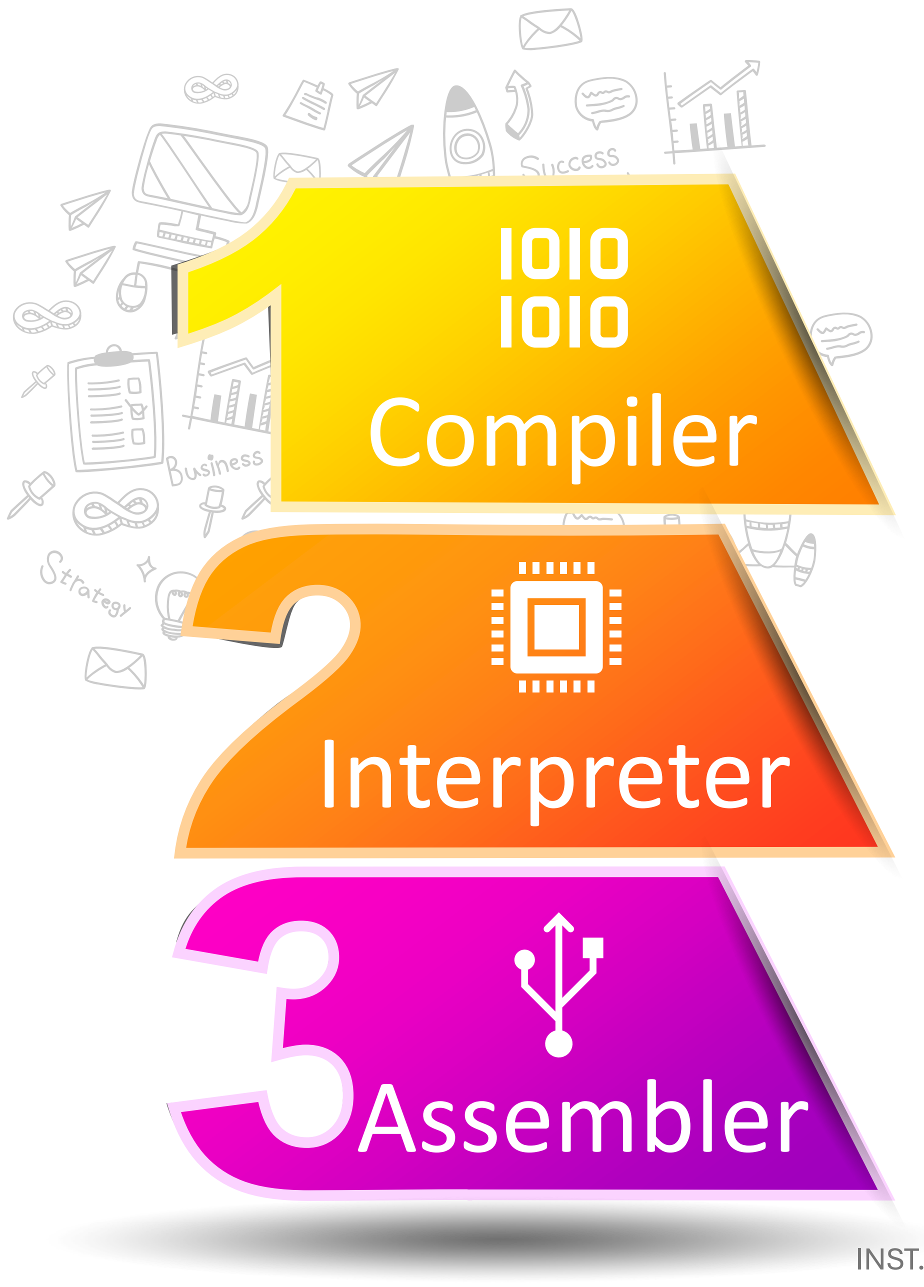
2) When the client requests a service from the server, the random port is opened for a very limited period and then closed again.


Request
One page
Client Side
24 Hour
Boot Server
www.bau.edu.jo
www.bau.edu.jo
Data → Request
← Page Response
Listening HTTP
Random
Application Layer
Transport Layer
50000
51000
68
67
21
22
80
443
21
22
80
443
Transport Layer
Application Layer
Segment
src port 51000
dest port 80
Data Request
Transport Layer
Application Layer
src port 80
dest port 51000
Data Page



Request Multiple pages
Client Side
24 Hour
Server Side
www.bau.edu.jo
www.bau_students.com
www.hu.edu.jo
Data → Request
www.bau.edu.jo
www.bau_students.com
Listening HTTPS
Listening HTTP
Application Layer
Transport Layer
50000
51000
52000
67
21
22
80
443
21
22
443
80
Multiplexing
De-Multiplexing
52000 443 www.hu.edu.jo
51000 80 www.bau_students.com
50000 443 www.bau.edu.jo
50000 443 www.bau.edu.jo
51000 80 www.bau_students.com
One Channel
Network Layer
One Channel
INST. : ENG.ALI BANI BAKAR & ENG


Request Multiple pages
Client Side
Server Side
24 Hour
www.bau.edu.jo
www.bau_students.com
www.hu.edu.jo
www.bau.edu.jo
www.bau_students.com
Page Response
Application Layer
Transport Layer
Network Layer
Listening HTTPS
Listening HTTP
50000
51000
52000
67
21
22
80
443
21
22
443
80
De-Multiplexing
Multiplexing
443 | 50000 | www.bau.edu.jo
80 | 51000 | www.bau_students.com
443 | 52000 | www.hu.edu.jo
80 | 51000 | www.bau_students.com
443 | 50000 | www.bau.edu.jo
One Channel
One Channel



It takes all instructions and executes them at once.

C Java

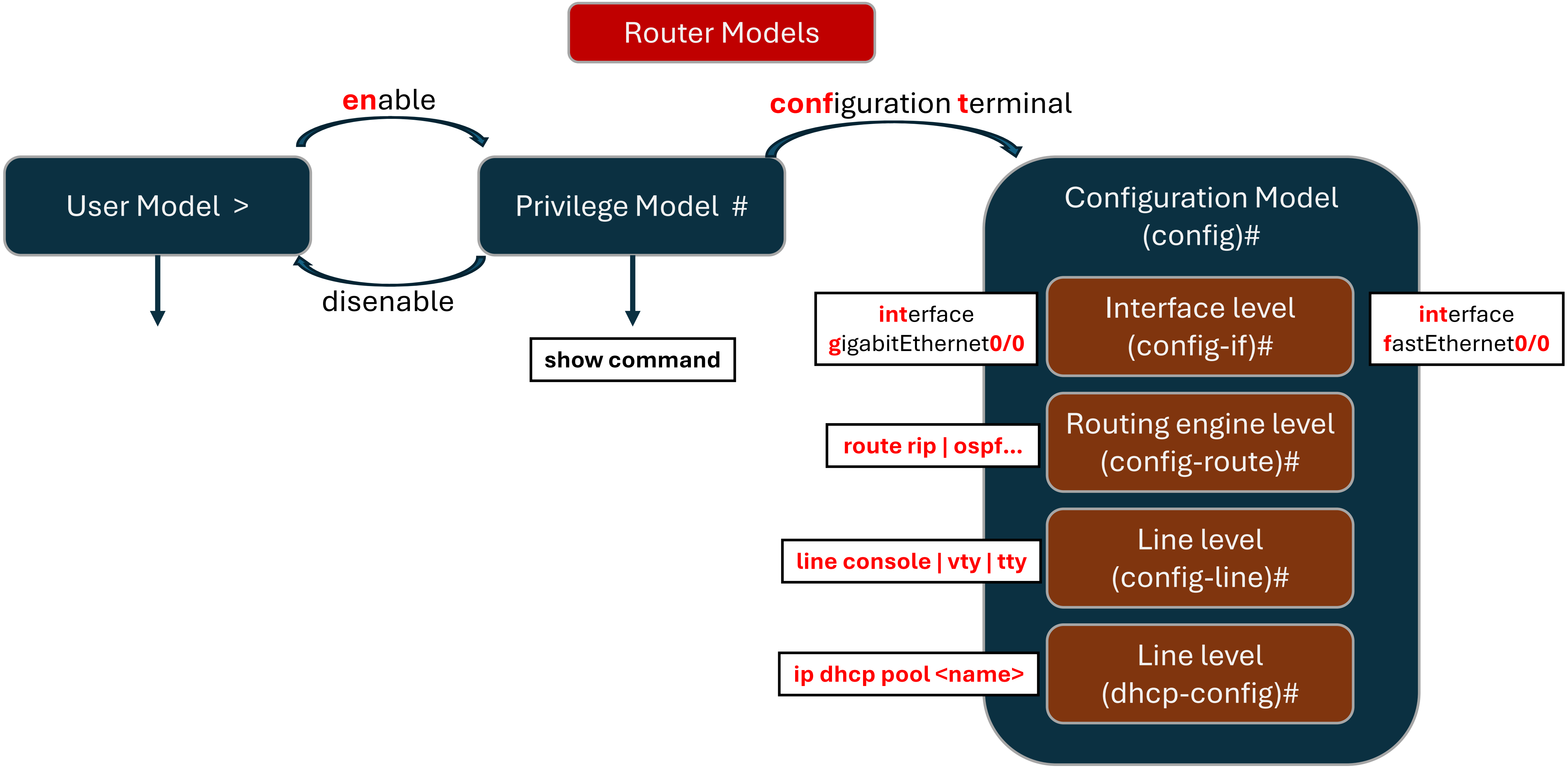
Instructions are executed line by line.

Gives direct output per line

CLI Linux MATLAB Python

Assembly

Basic configuration for Router using CLI



Route Command

Router>**en**able

Router#**confi**gure **t**erminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#**int**erface **g**igabitEthernet**0/0**

Router(config-if)#**ip add**ress **192.168.0.1 255.255.255.0**

Router(config-if)#**no sh**utdown

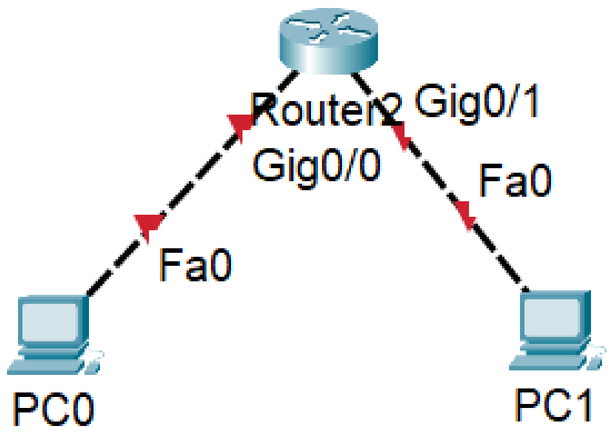
Router(config-if)#**exit**

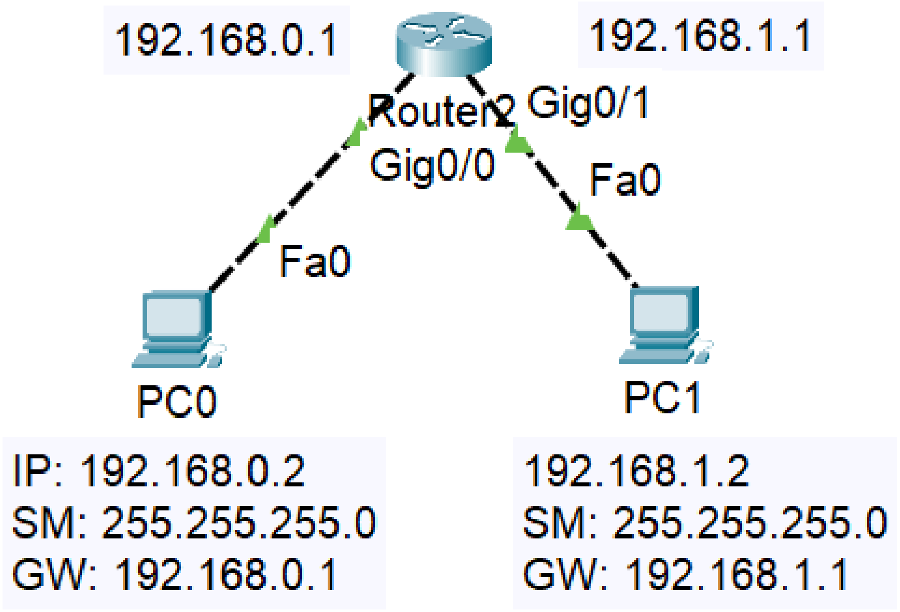
Router(config)#**int**erface **g**igabitEthernet**0/1**

Router(config-if)#**ip add**ress **192.168.1.1 255.255.255.0**

Router(config-if)#**no sh**utdown

Router(config-if)#**do wr**ite





PC in CLI

ipconfig <ip address> <subnet mask> <default gateway>

PC0:

C:\>**ipconfig 192.168.0.2 255.255.255.0 192.168.0.1**

PC1:

C:\>**ipconfig 192.168.1.2 255.255.255.0 192.168.1.1**

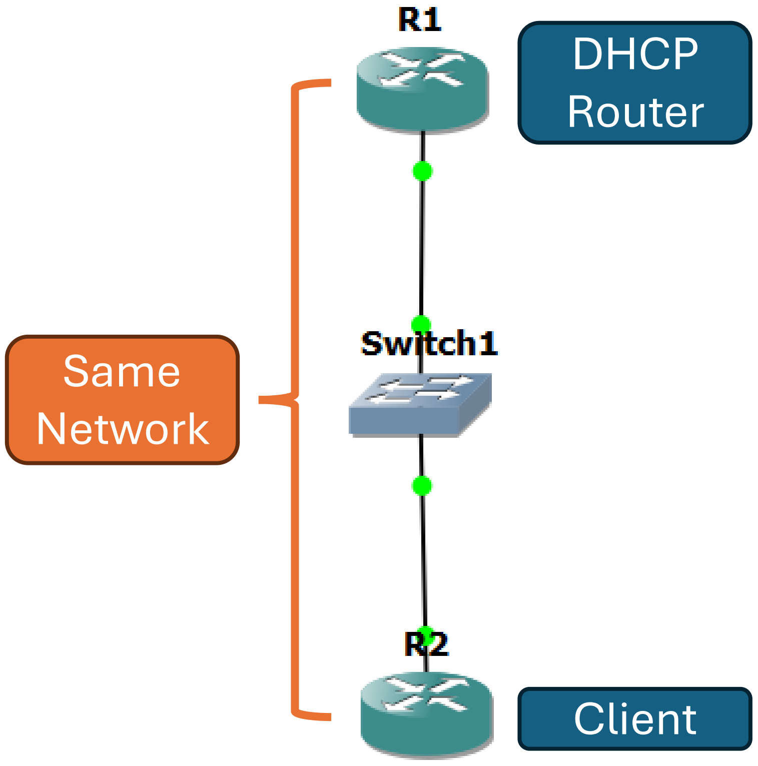
Ping:

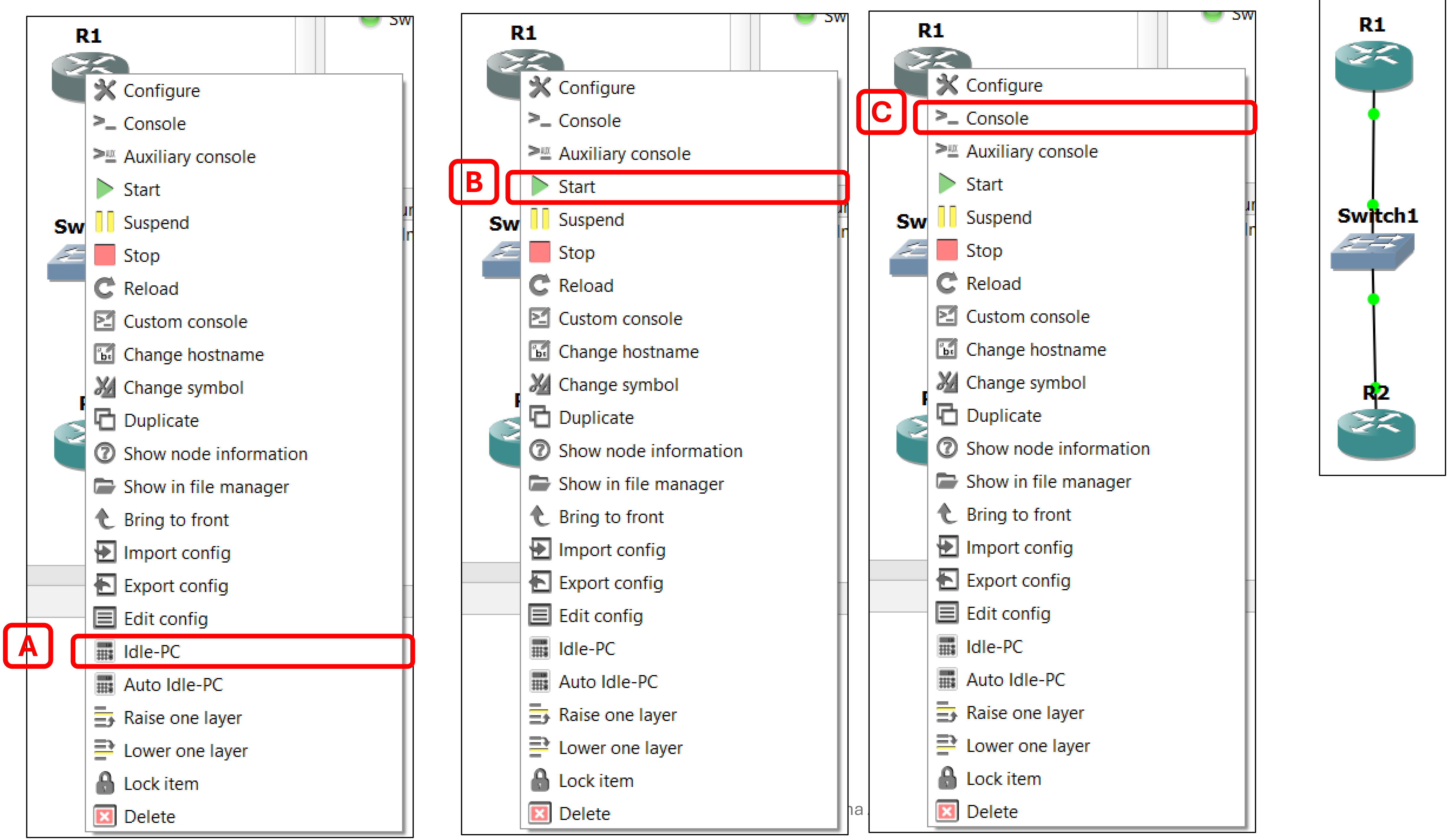
PC0:

ping 192.168.1.2

DHCP Router

- Q: DHCP operates at the Application Layer (Layer 7), and routers are network devices that primarily function at the Network Layer (Layer 3). How can a router also serve as a DHCP server, which is typically a Layer 7 function?
- A: **Integrated Services**: Modern routers often come with integrated software that includes DHCP server functionality. This allows them to manage IP address allocation and network configuration for devices on the network.




R1
Configure
Console
Auxiliary console
Start
Suspend
Stop
Reload
Custom console
Change hostname
Change symbol
Duplicate
Show node information
Show in file manager
Bring to front
Import config
Export config
Edit config
Idle-PC
Auto Idle-PC
Raise one layer
Lower one layer
Lock item
Delete
A
R1
Configure
Console
Auxiliary console
Start
Suspend
Stop
Reload
Custom console
Change hostname
Change symbol
Duplicate
Show node information
Show in file manager
Bring to front
Import config
Export config
Edit config
Idle-PC
Auto Idle-PC
Raise one layer
Lower one layer
Lock item
Delete
B
R1
Configure
Console
Auxiliary console
Start
Suspend
Stop
Reload
Custom console
Change hostname
Change symbol
Duplicate
Show node information
Show in file manager
Bring to front
Import config
Export config
Edit config
Idle-PC
Auto Idle-PC
Raise one layer
Lower one layer
Lock item
Delete
C
R1
Switch1
R2

R1 DHCP Router

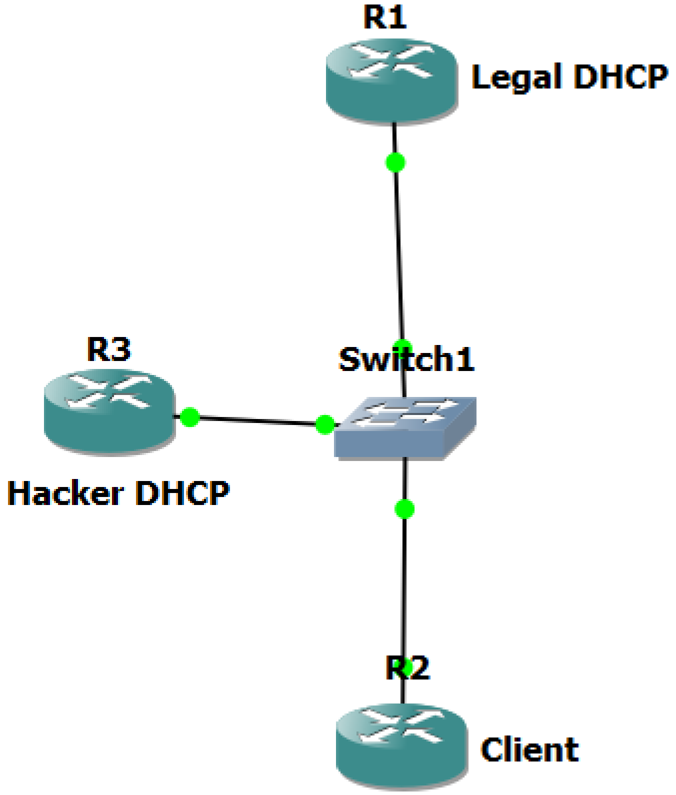
```
R1#configure terminal
R1(config)#interface fastEthernet0/0
R1(config-if)#ip address 192.168.0.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit

R1#configure terminal
R1(config)#ip dhcp pool legal
R1(dhcp-config)#network 192.168.0.0 255.255.255.0
R1(dhcp-config)#dns-server 1.1.1.1
R1(dhcp-config)#do write
Building configuration...
[OK]
```

R2 Client

```
R2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface fastEthernet0/0
R2(config-if)#ip address dhcp
R2(config-if)#no shutdown
R2(config-if)#exit

R5#show ip interface
FastEthernet0/0 is up, line protocol is up
  Internet address is 192.168.0.2/24
  Broadcast address is 255.255.255.255
  Address determined by DHCP
```

Adding Hacker DHCP Router ➔ who will win?

```
R3#configure terminal
R3(config)#interface fastEthernet0/0
R3(config-if)#ip address 172.16.0.1   255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit

R3#configure terminal
R3(config)#ip dhcp pool legal
R3(dhcp-config)#network 172.16.0.0   255.255.255.0
R3(dhcp-config)#dns-server 1.1.1.1
R3(dhcp-config)#do write
Building configuration...
[OK]
```



How win? Legal or Hacker

```
R2(config-if)#ip add dhcp
R2(config-if)#
*Jul 19 07:48:59.395: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Jul 19 07:49:00.395: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config-if)#
*Jul 19 07:49:09.911: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 172.16.0.2, mask 255.255.0.
0, hostname R2

R2(config-if)#ip add dhcp
R2(config-if)#
*Jul 19 07:50:00.539: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 192.168.0.2, mask 255.255.0.0, hostname R2

R2(config-if)#ip add dhcp
R2(config-if)#
*Jul 19 07:50:34.147: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 172.16.0.3, mask 255.255.0.0, hostname R2
```

Random

dhcp

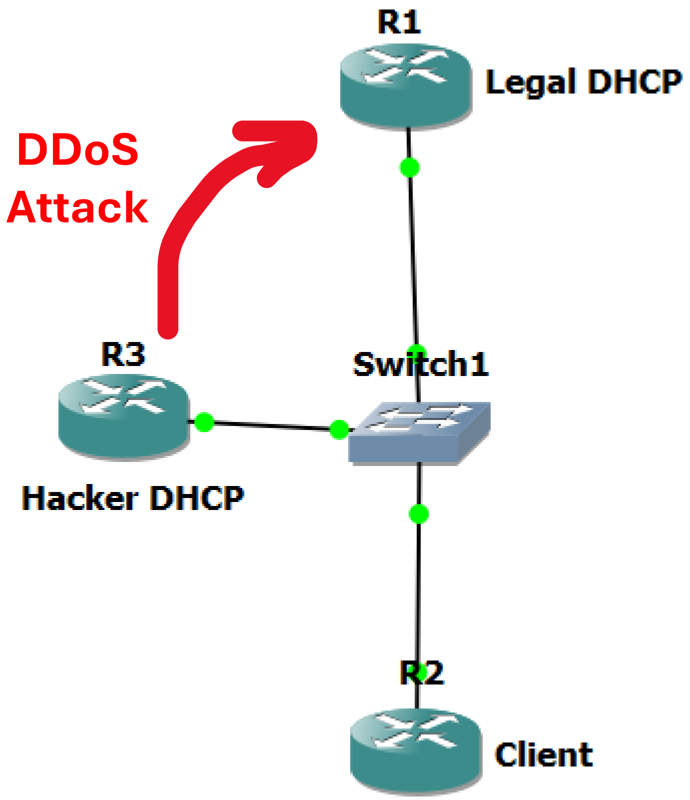
No.	Time	Source	Destination	Protocol	Length	Info
9	80.102644	0.0.0.0	255.255.255.255	DHCP	333	DHCP Discover - Transaction ID 0x2584
12	81.840925	172.16.0.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x2584
13	81.902026	0.0.0.0 Hacker	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0x2584
14	81.933479	172.16.0.1 win	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x2584
16	82.384133	192.168.0.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x2584
27	132.273116	172.16.0.2	172.16.0.1	DHCP	321	DHCP Release - Transaction ID 0x2584
30	137.562315	172.16.0.2	172.16.0.1	DHCP	321	DHCP Release - Transaction ID 0x2584
32	143.437290	0.0.0.0	255.255.255.255	DHCP	333	DHCP Discover - Transaction ID 0x114
34	143.452849	192.168.0.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x114
35	143.544879	0.0.0.0 Lega	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0x114
36	143.561048	192.168.0.1 win	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x114
39	144.865556	172.16.0.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x114
48	175.717796	192.168.0.2	192.168.0.1	DHCP	321	DHCP Release - Transaction ID 0x114
49	180.897214	192.168.0.2	192.168.0.1	DHCP	321	DHCP Release - Transaction ID 0x114
51	186.256325	0.0.0.0	255.255.255.255	DHCP	333	DHCP Discover - Transaction ID 0x1983
52	186.272151	172.16.0.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x1983
54	186.459056	0.0.0.0 Hacker	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0x1983
55	186.475138	172.16.0.1 win	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x1983
57	188.262781	192.168.0.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x1983

How can Hacker win?

Winning and losing in this is **random**.

How can a hacker make his win rate higher than a legitimate router?

The hacker can use some network tactics to increase his chance of winning. For example, he can make the **legitimate router busy** in other work, such as the hacker sending a **DDoS attack** on the router.



SubNetting

- Q: Why we need to use subnetting?
- Subnetting allows for the creation of multiple subnets, enhancing network organization, management, and security.
- Easy to use and manage.

Ex1:

IP: 192.168.0.4/24

Subnet mask: 255.255.255.0

3 Labs

➔

Subnet mask: 255.255.255.**11**000000

3 labs ➔ 2 Bit needs

IP1: 192.168.0.**00**000000**/26**

IP1: 192.168.0.**0/26**

Subnet mask: 255.255.255.**11**000000

Network ID: 192.168.0.00000000 ➔ 0

Broadcast: 192.168.0.00111111 ➔ 63

IP2: 192.168.0.**01**000000/26

IP2: 192.168.0.**64/26**

Subnet mask: 255.255.255.**11**000000

Network ID: 192.168.0.01000000 ➔ 64

Broadcast: 192.168.0.01111111 ➔ 127

IP3: 192.168.0.**10**000000**/26**

IP3: 192.168.0.**128/26**

Subnet mask: 255.255.255.**11**000000

Network ID:192.168.0.10000000 ➔ 128

Broadcast: 192.168.0.10111111 ➔ 191

IP4: 192.168.0.**11**000000**/26**

IP4: 192.168.0.**192/26**

Subnet mask: 255.255.255.**11**000000

Network ID:192.168.0.11000000 ➔ 192

Broadcast: 192.168.0.11111111 ➔ 225

Ex2:

IP: 192.168.5.0/24

Subnet mask: 255.255.255.0

2 Labs

→

Subnet mask: 255.255.255.**1**0000000

2 labs → 1 Bit need

IP1: 192.168.5.**0**0000000**/25**

IP1: 192.168.5.**0/25**

Subnet mask: 255.255.255.**1**0000000

Network ID: 192.168.5.00000000 → 0

Broadcast: 192.168.5.01111111 → 127

IP2: 192.168.5.**1**0000000**/25**

IP2: 192.168.5.**128/25**

Subnet mask: 255.255.255.**1**0000000

Network ID:192.168.5.10000000 → 128

Broadcast: 192.168.5.11111111 → 255

Ex3:

IP: 192.168.5.0/24

Subnet mask: 255.255.255.0 → 255.255.255.11100000

7 Labs → 3 bits

IP1: 192.168.5.00000000/27

IP1: 192.168.5.0/27

Subnet mask: 255.255.255.11100000

Network ID: 192.168.5.00000000 → 0

Broadcast: 192.168.5.00011111 → 31

IP2: 192.168.5.00100000/27

IP2: 192.168.5.32/27

Subnet mask: 255.255.255.11100000

Network ID: 192.168.5.00100000 → 32

Broadcast: 192.168.5.00111111 → 63

IP3: 192.168.5.01000000/27

IP3: 192.168.5.64/27

Subnet mask: 255.255.255.11100000

Network ID: 192.168.5.01000000 → 64

Broadcast: 192.168.5.01011111 → 95

IP4: 192.168.5.01100000/27

IP4: 192.168.5.96/27

Subnet mask: 255.255.255.11100000

Network ID: 192.168.5.011000000 → 96

Broadcast: 192.168.5.01111111 → 127

IP5: 192.168.5.10000000/27

IP5: 192.168.5.128/27

Subnet mask: 255.255.255.11100000

Network ID: 192.168.5.10000000 → 128

Broadcast: 192.168.5.10011111 → 159

IP6: 192.168.5.10100000/27

IP6: 192.168.5.160/27

Subnet mask: 255.255.255.11100000

Network ID: 192.168.5.10100000 → 160

Broadcast: 192.168.5.10111111 → 191

IP7: 192.168.5.11000000/27

IP7: 192.168.5.192/27

Subnet mask: 255.255.255.11100000

Network ID: 192.168.5.11000000 → 192

Broadcast: 192.168.5.11011111 → 223

IP8: 192.168.5.11100000/27

IP8: 192.168.5.224/27

Subnet mask: 255.255.255.11100000

Network ID: 192.168.5.11100000 → 224

Broadcast: 192.168.5.11111111 → 255

Day 4

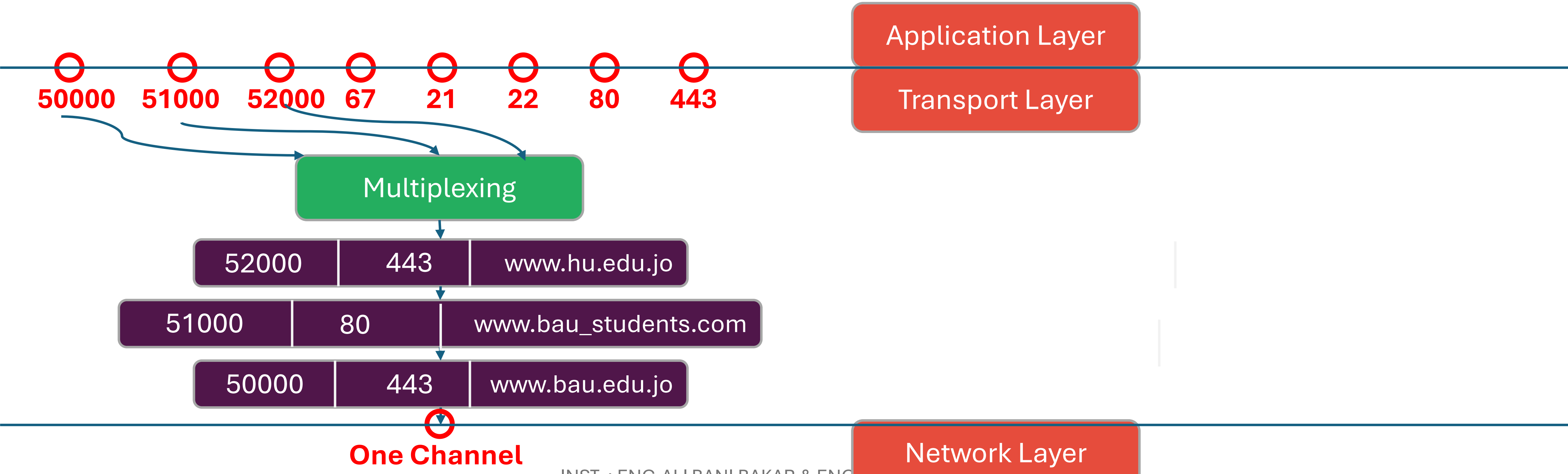
- Outline
  - Transport Layer Function
    - Segmentation
    - Flow Control
    - Congestion Control
    - Error Detection
  - Transport Layer Protocol
    - TCP
    - UDP
  - TCP Session

Day 4

- Outline Continue…
  - Router Password
    - Enable Password
    - Enable Secret
    - Console Password
    - vty Password
  - Telnet in GNS3 & Wireshark
  - Public & Private IP
  - Application Protocols
  - Intro to VLAN

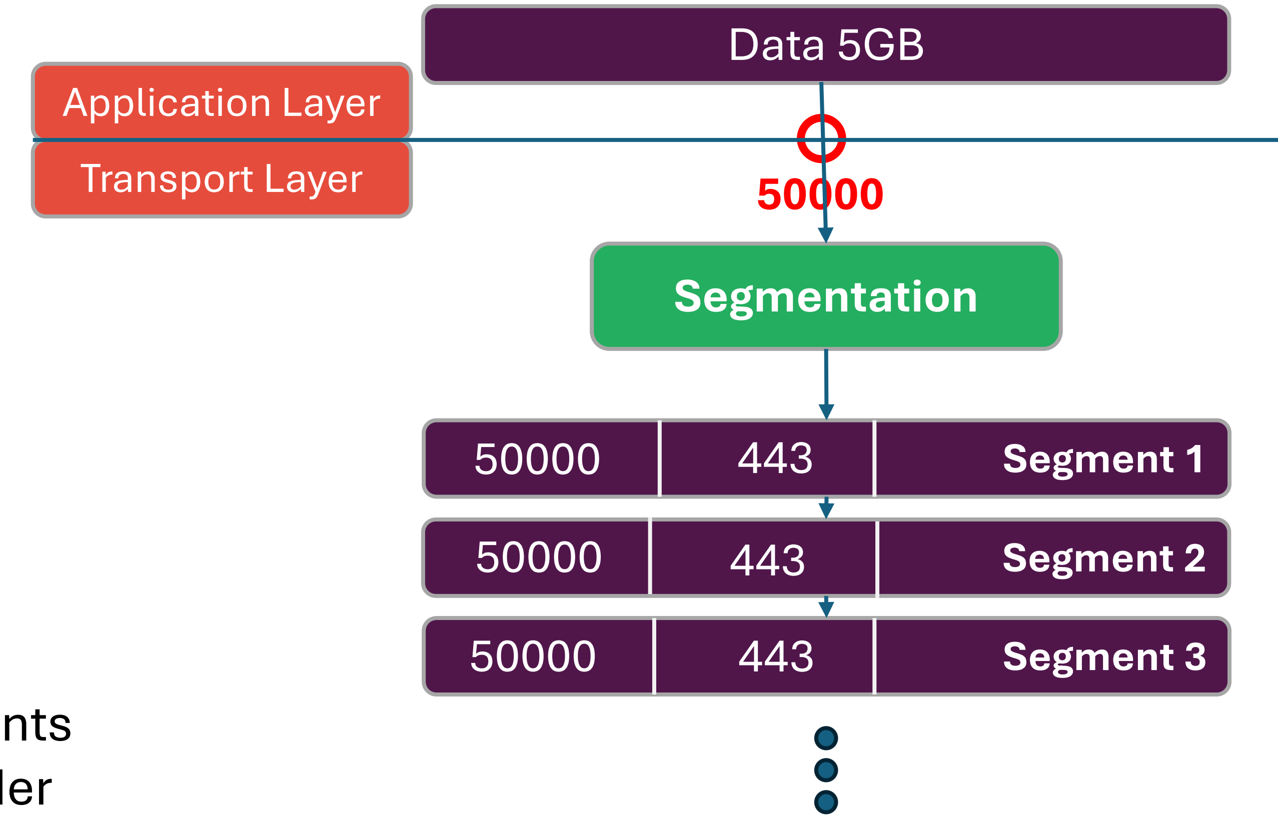
Transport Layer Function

1. **Multiplexing:** combine multiple signals into one medium or channel, from multiple ports coming from Application layer to one channel to Network layer.



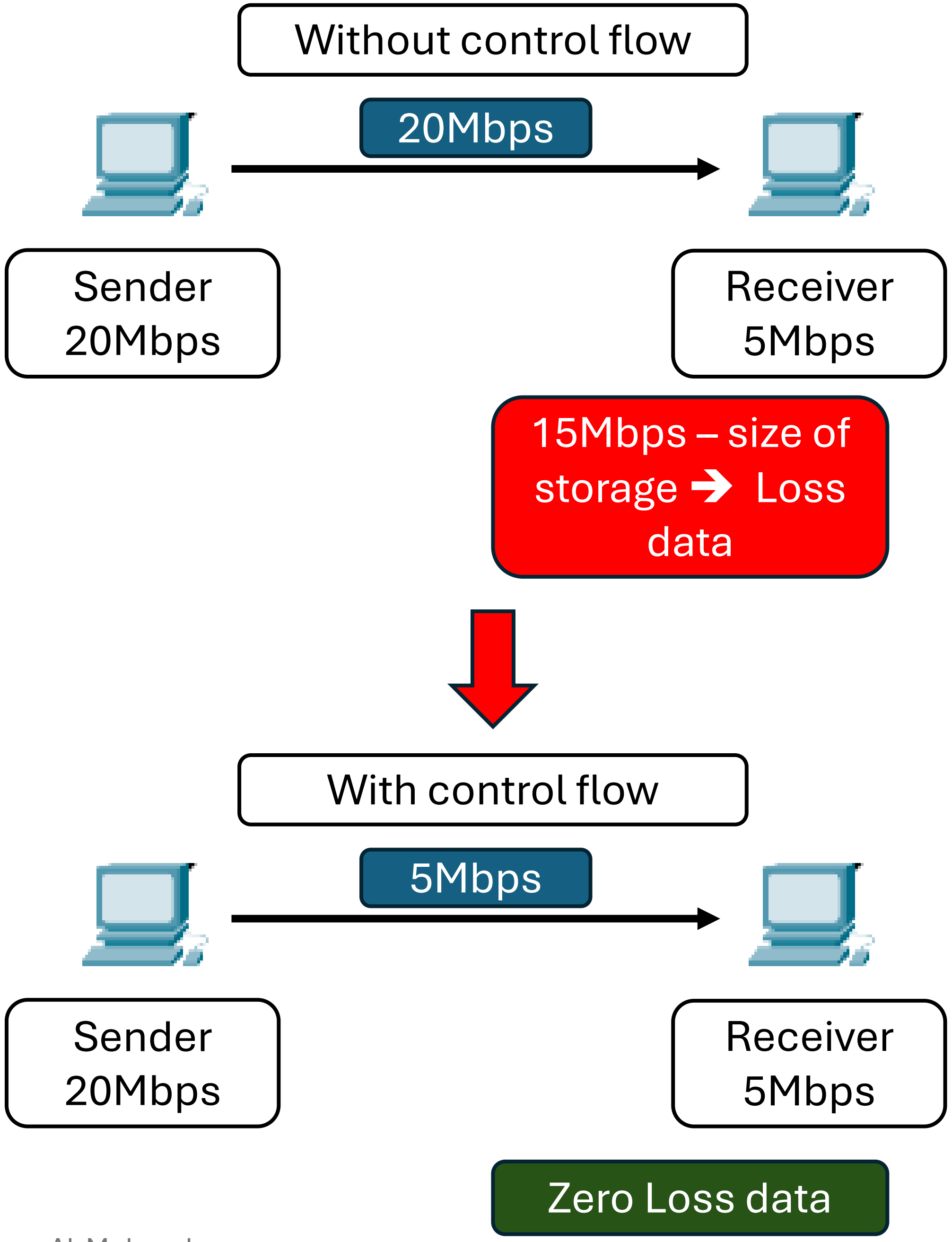
2. Segmentation (Both)

- Dividing data into smaller, more manageable pieces for transmission over a network has several advantages:
- **Data Integrity**: Smaller segments reduce the likelihood of errors.
- **Error Reduction**: Transmitting smaller amounts of data at a time decreases the chances of errors occurring.
- **Efficient Routing:** Each segment can take its own path through the network, depending on traffic conditions.
- **Reordering by Receiver:** The order in which segments are received is not important; the receiver will reorder the segments into the correct sequence, It make at:
  - TCP ➔ Transport Layer
  - UDP ➔ Application Layer



3. Flow Control (TCP Only)

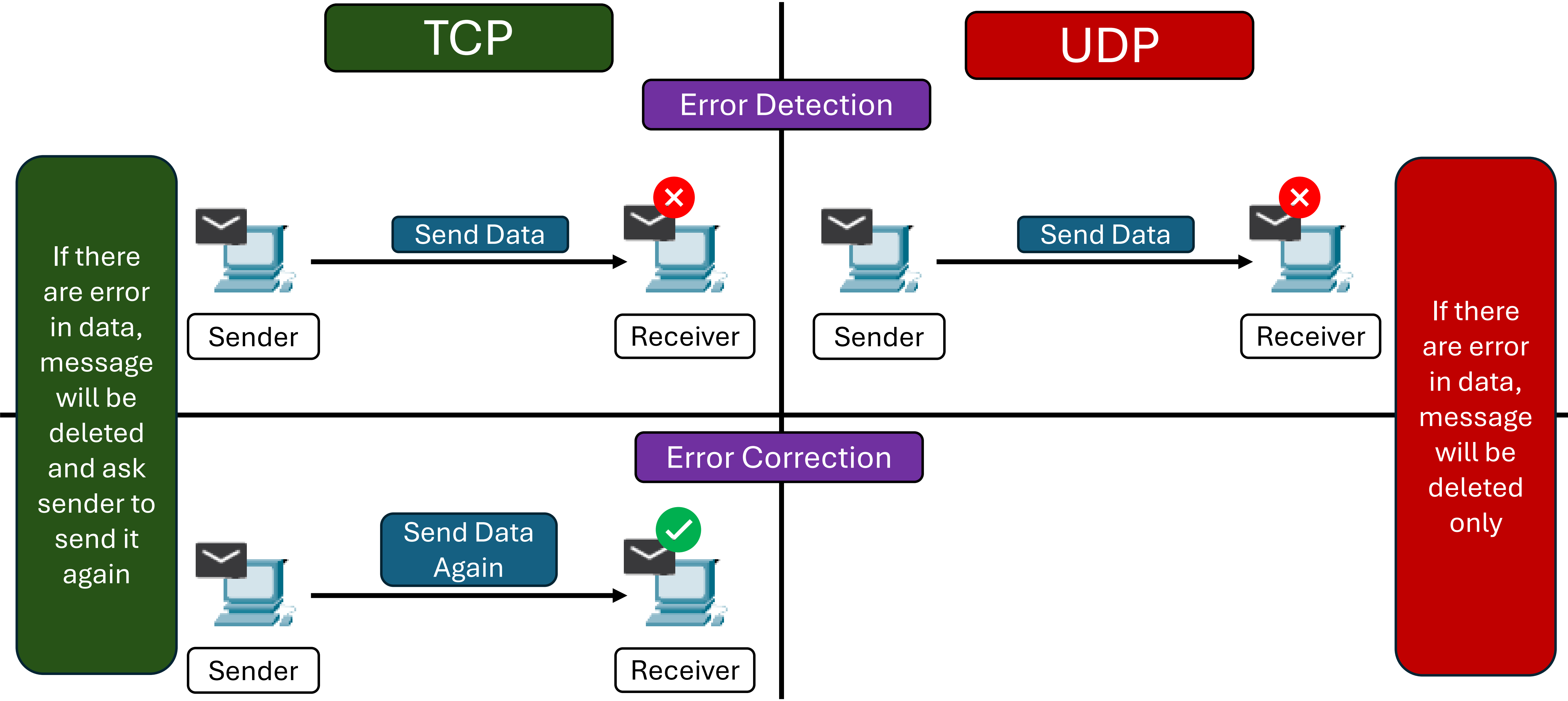
- manage the rate of data transmission between a sender and a receiver, ensuring that the sender does not overwhelm the receiver with too much data at once.
- **Problem:** If the recipient of the message is slow and cannot receive a large amount of data in a short time, and the sender is fast and can handle a large amount of data by sending and receiving, then the receiver loses some data.
- **Solution:** Before the data is sent, the sender and the receiver agree on the speed of sending the data, depending on how much the slower one can tolerate.
- If the route allows more data to be sent, it will be done



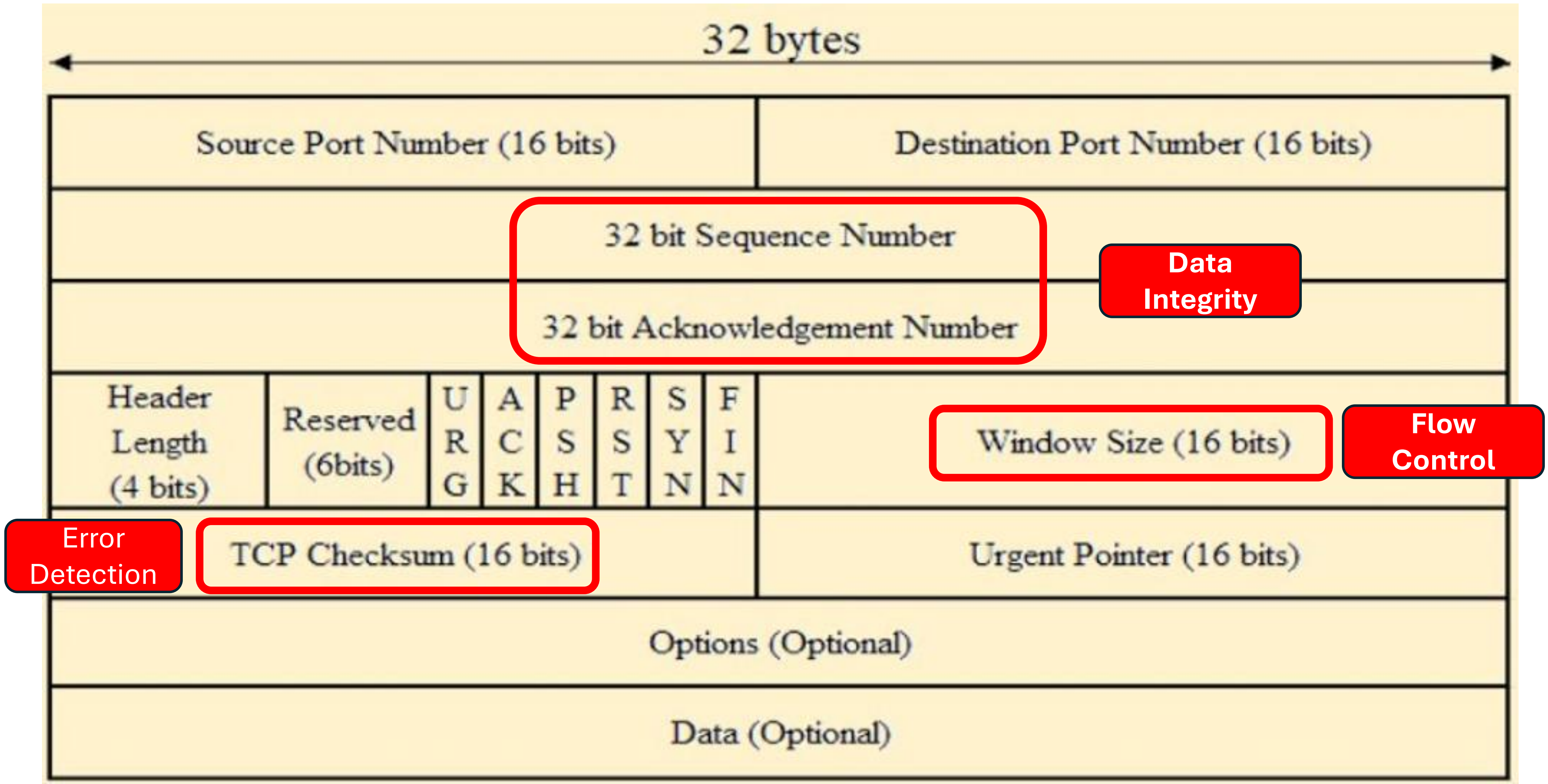
4. **Congestion Control** **(TCP Only)**

- manage the volume of data being sent into the network to avoid overwhelming it. This involves dynamically by controlling the window size adjusting the rate of data transmission based on current network conditions.

5. Error Detection & Correction

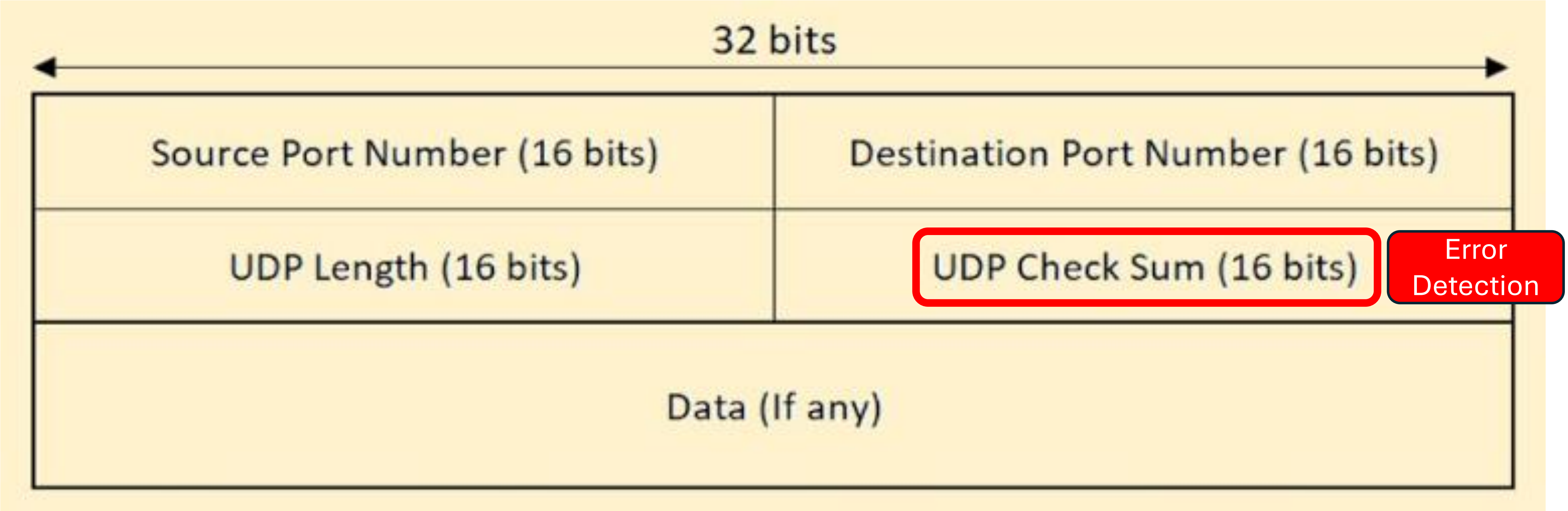


TCP Header



32 bytes
Source Port Number (16 bits)
Destination Port Number (16 bits)
32 bit Sequence Number
32 bit Acknowledgement Number
Data Integrity
Header Length (4 bits)
Reserved (6bits)
URG
ACK
PSH
RST
SYN
FIN
Window Size (16 bits)
Flow Control
Error Detection
TCP Checksum (16 bits)
Urgent Pointer (16 bits)
Options (Optional)
Data (Optional)

UDP Header


32 bits
Source Port Number (16 bits)
Destination Port Number (16 bits)
UDP Length (16 bits)
UDP Check Sum (16 bits)
Error Detection
Data (If any)

	TCP	UDP
Connection	✓	✗
Data Reliable	✓	✗
Header	big	small
Error Detection	✓	✓
Error Correction	✓	✗
Performance	✗	✓

Application Layer Protocol → (TCP | UDP)

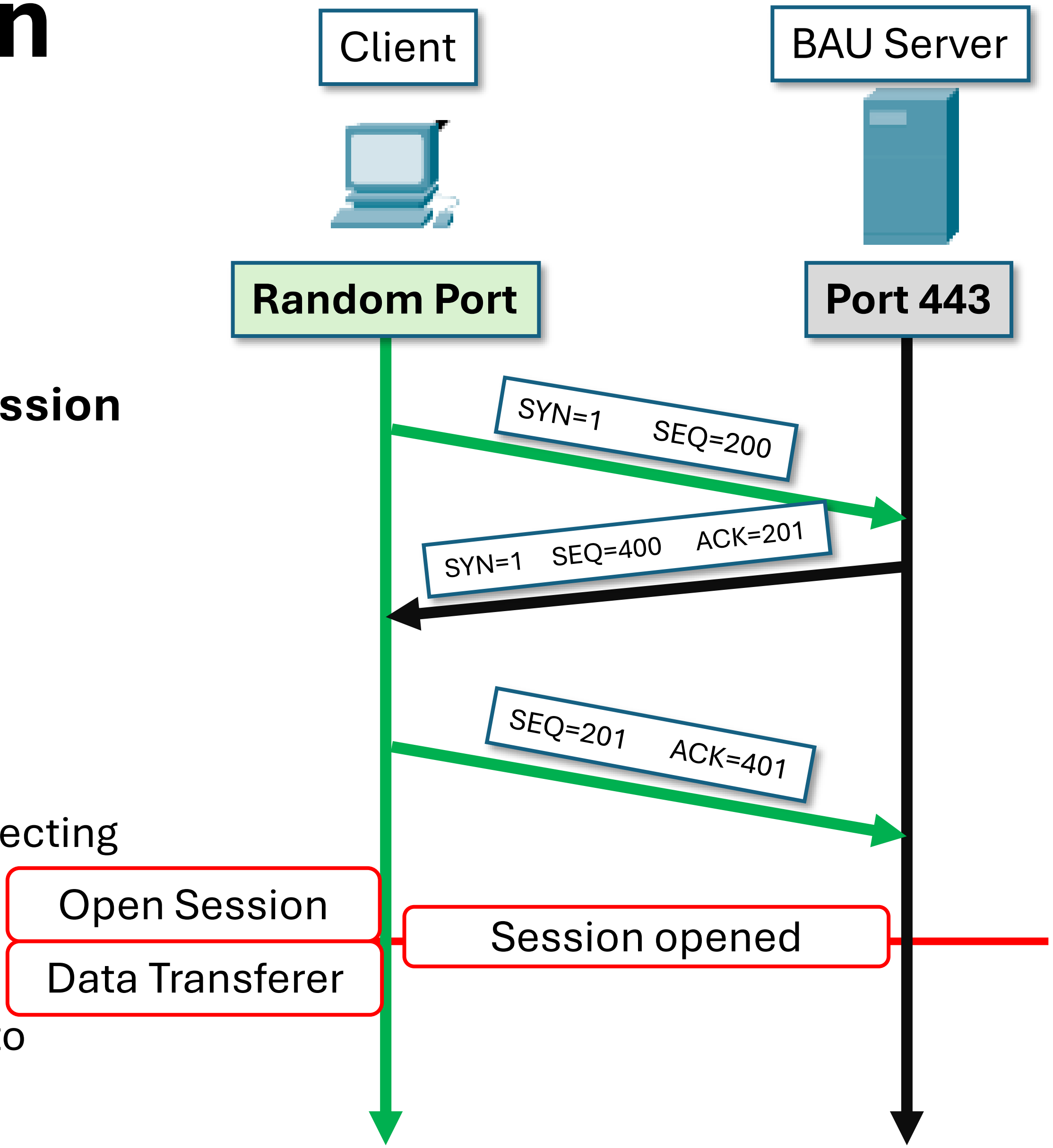
	(TCP | UDP)
FTP (20)	TCP
HTTP (80) & HTTPS (443)	TCP
SMTP (25) & POP3 (110)	TCP
Telnet (23) & SSH (22)	TCP
DNS (53)	Primarily UDP for queries, TCP for zone transfers
DHCP (67 (server), 68 (client))	UDP
TFTP (69)	UDP
ICMP (??)	??

Session

1. Open Session (3-way handshaking) [Hi].
2. Data Transfer [Data].
3. Terminate Session (4-way handshaking) [Bye].

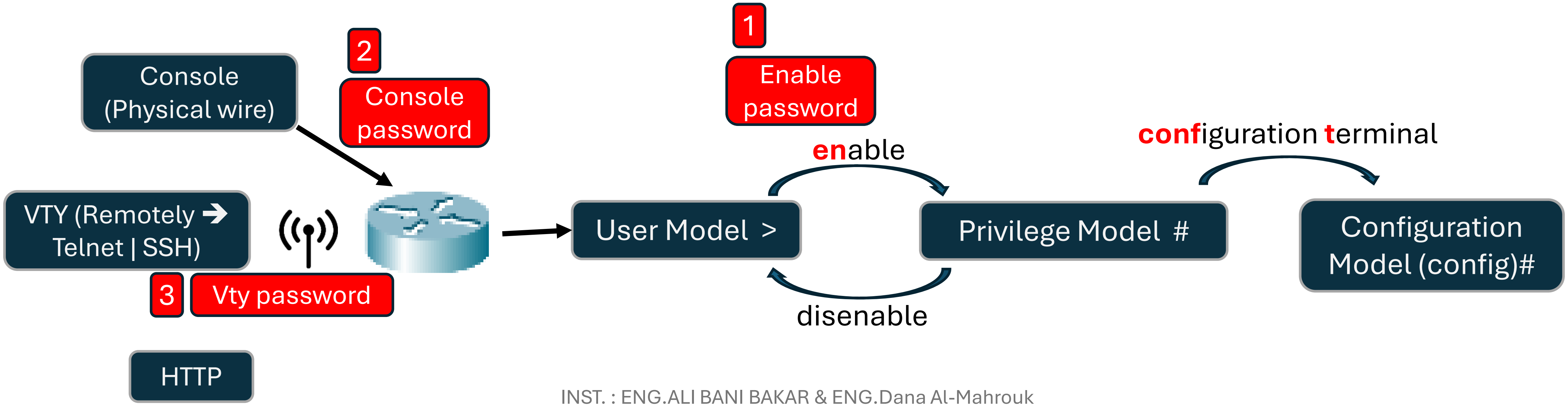
TCP/IP ➔ Transport Layer ➔ TCP ➔ Session ➔ Open Session

- SYN
  - 1: Open Session
  - 0: else
- SEQ (Sequence Number)
- ensuring data is delivered in the correct order and assists in detecting missing segments.
- ACK (Acknowledgment Number)
- used by the receiver to indicate the next byte of data it expects to receive from the sender.



Router Password

- Types:
  1. Enable Password, Secret Password
  2. Console Password
  3. Line vty Password



1. Enable Password

R>**en**able

R#**confi**gure **t**erminal

R(config)#**enable password** 1234

R(config)#do write

R(config)#exit

R>enable

Password: 1234

R#show run

2. Secret Password

R>**en**able

R#**confi**gure **t**erminal

R(config)#**enable secret** abcd

R(config)#do write

R(config)#exit

R>enable

Password: 1234

R#show run

Hash Value → MD5

3. Console Password

R>**en**able

R#**confi**gure **t**erminal

R(config)#**line console 0**

R(config-**line**)#**password** zinc

R(config-**line**)#**login**

R(config-**line**)#do write

Turn Off Router

Turn On Router

Open Console

Password: zinc

R>

4. vty Password

R>**en**able

R#**confi**gure **t**erminal

R(config)#**line vty 0 3**

R(config-**line**)#**password** zinc

R(config-**line**)#**login**

R(config-**line**)#do write

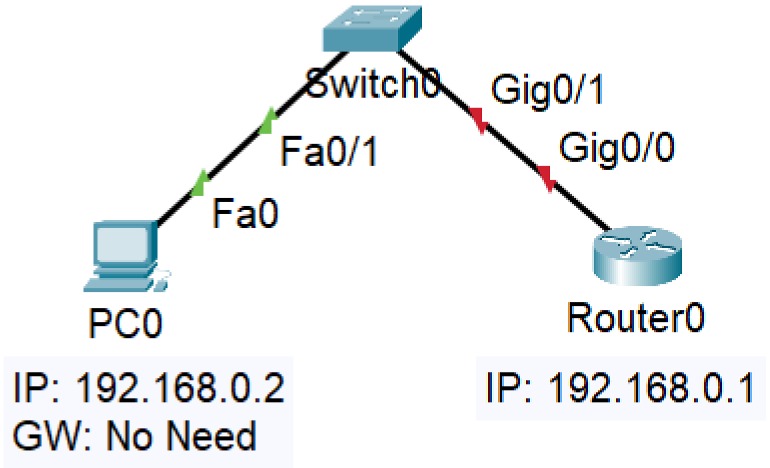
Turn Off Router

Turn On Router

Open using Telnet | SSH

VTY Connection Router using Telnet

```
R>enable
R#configure terminal
R(config)#interface GigaEthernet0/0
R(config)#ip address 192.168.0.1 255.255.255.0
R(config)#no shutdown
R(config)#do ping 192.168.0.2
R(config)#exit
R(config)#line vty 0 4
R(config-vty)#password zinc
R(config-vty)#login
R(config-vty)#exit
R(config)#enable password 1234
R(config)#do write
```




```
C:/>telnet 192.168.0.1
Trying ..... Open
Password: zinc
R>enable
Password: 1234
R#
```

Secure of VTY Connection in GNS3

R1#conf t

R1(config)#int f0/0

R1(config-if)#ip add 192.168.0.1 255.255.255.0

R1(config-if)#no shutdown

R1(config-if)#exit

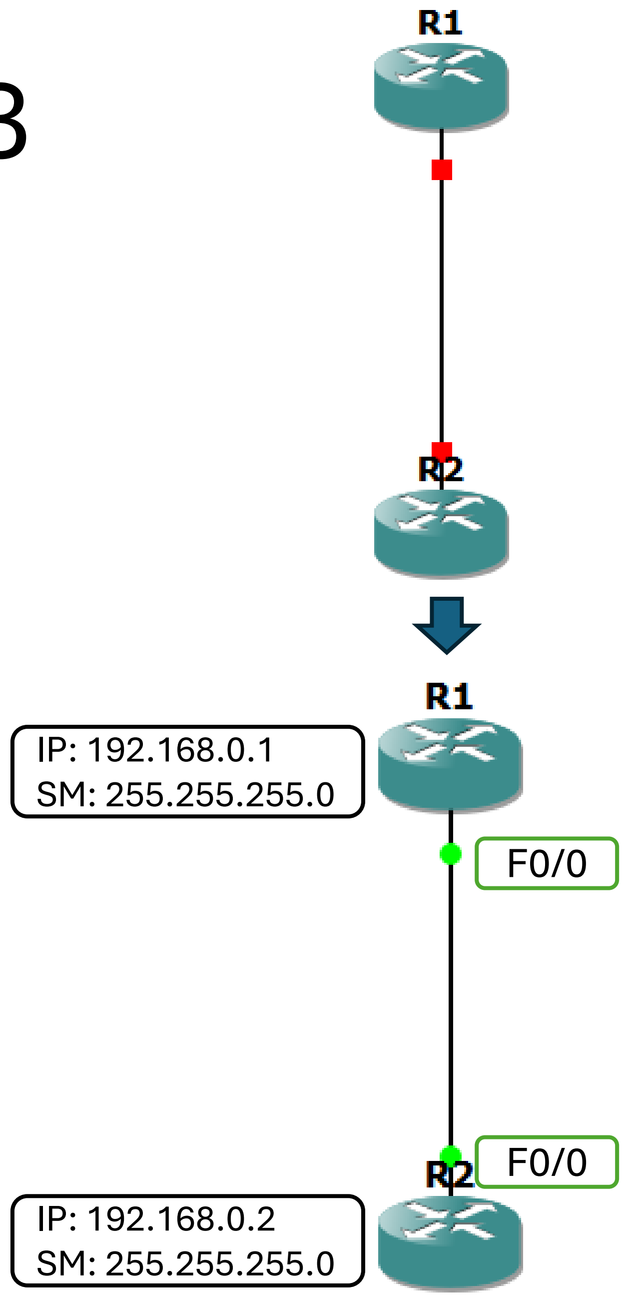
R2#conf t

R2(config)#int f0/0

R2(config-if)#ip add 192.168.0.2 255.255.255.0

R2(config-if)#no shutdown

R1(config-if)#exit



```
R1#ping 192.168.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.0.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/57/80 ms
R1#conf t
R1(config)#enable secret 1234
R1(config)#line vty 0 4
R1(config-line)#password zinc
R1(config-line)#login
R1(config-line)#do wr
```

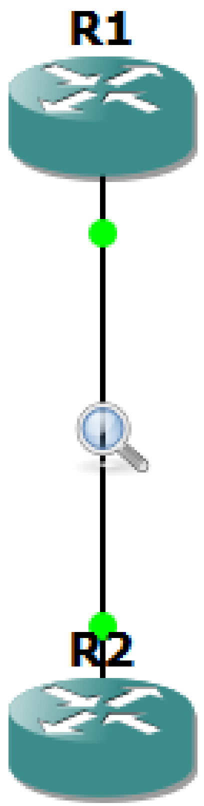
```
R2#telnet 192.168.0.1
Trying 192.168.0.1 ... Open
User Access Verification
Password: <zinc>
R1>en
Password: <1234>
R1#
```

Telnet Capture

Q: What can Hacker see?

Q: How is client and how is server?

Q: Telnet Layer?



*- [R1 FastEthernet0/0 to R2 FastEthernet0/0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
16	58.991851	192.168.0.2	192.168.0.1	TCP	60	41176 → 23 [SYN] Seq=0 Win=4128 Len=0 MSS=1460
17	59.069678	192.168.0.1	192.168.0.2	TCP	60	23 → 41176 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=1460
18	59.085102	192.168.0.2	192.168.0.1	TCP	60	41176 → 23 [ACK] Seq=1 Ack=1 Win=4128 Len=0
19	59.085102	192.168.0.2	192.168.0.1	TELNET	63	Telnet Data ...
20	59.085102	192.168.0.2	192.168.0.1	TCP	60	[TCP Dup ACK 18#1] 41176 → 23 [ACK] Seq=10 Ack=1 Win=4128 Len=0
21	59.114983	192.168.0.1	192.168.0.2	TCP	60	23 → 41176 [ACK] Seq=1 Ack=10 Win=4119 Len=0
22	59.114983	192.168.0.1	192.168.0.2	TELNET	66	Telnet Data ...
23	59.130740	192.168.0.1	192.168.0.2	TELNET	96	Telnet Data ...
24	59.130740	192.168.0.1	192.168.0.2	TELNET	60	Telnet Data ...
25	59.130740	192.168.0.1	192.168.0.2	TELNET	60	Telnet Data ...
26	59.130740	192.168.0.2	192.168.0.1	TCP	60	41176 → 23 [ACK] Seq=10 Ack=13 Win=4116 Len=0
27	59.130740	192.168.0.2	192.168.0.1	TELNET	60	Telnet Data ...
28	59.130740	192.168.0.2	192.168.0.1	TELNET	60	Telnet Data ...
29	59.130740	192.168.0.2	192.168.0.1	TELNET	63	Telnet Data ...
30	59.145948	192.168.0.1	192.168.0.2	TCP	60	23 → 41176 [ACK] Seq=64 Ack=16 Win=4113 Len=0
31	59.145948	192.168.0.1	192.168.0.2	TELNET	60	Telnet Data ...
32	59.145948	192.168.0.2	192.168.0.1	TCP	60	41176 → 23 [ACK] Seq=25 Ack=58 Win=4071 Len=0
33	59.161356	192.168.0.2	192.168.0.1	TCP	60	41176 → 23 [ACK] Seq=25 Ack=67 Win=4062 Len=0
35	61.857253	192.168.0.2	192.168.0.1	TELNET	60	Telnet Data ...
36	62.060273	192.168.0.1	192.168.0.2	TCP	60	23 → 41176 [ACK] Seq=67 Ack=26 Win=4103 Len=0
37	62.076220	192.168.0.2	192.168.0.1	TELNET	60	Telnet Data ...
39	62.322954	192.168.0.1	192.168.0.2	TCP	60	23 → 41176 [ACK] Seq=67 Ack=27 Win=4102 Len=0
40	62.570260	192.168.0.2	192.168.0.1	TELNET	60	Telnet Data ...
41	62.663879	192.168.0.2	192.168.0.1	TELNET	60	Telnet Data ...
42	62.679094	192.168.0.1	192.168.0.2	TCP	60	23 → 41176 [ACK] Seq=67 Ack=29 Win=4100 Len=0
43	63.048342	192.168.0.2	192.168.0.1	TELNET	60	Telnet Data ...
44	63.063353	192.168.0.1	192.168.0.2	TELNET	60	Telnet Data ...
45	63.265852	192.168.0.2	192.168.0.1	TCP	60	41176 → 23 [ACK] Seq=31 Ack=72 Win=4057 Len=0
46	64.855172	192.168.0.2	192.168.0.1	TELNET	60	Telnet Data ...

```
Frame 49: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
Ethernet II, Src: ca:01:32:18:00:00 (ca:01:32:18:00:00), Dst: ca:02:38:f0:00:00 (ca:02:38:f0:00:00)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2
Transmission Control Protocol, Src Port: 23, Dst Port: 41176, Seq: 73, Ack: 33, Len: 1
Telnet
```

```
0000  ca 02 38 f0 00 00 ca 01  32 18 00 00 08 00 45 c0   ··8·····2·····E·
0010  00 29 8a 23 00 00 ff 06  af 97 c0 a8 00 01 c0 a8   ·)·#············
0020  00 02 00 17 a0 d8 f0 cf  8d 30 5b 9e c4 6a 50 18   ·········0[··jP·
0030  10 00 71 7e 00 00 6e 00  00 00 00 00               ··q~··n·····
```

Transmission Control Protocol: Protocol

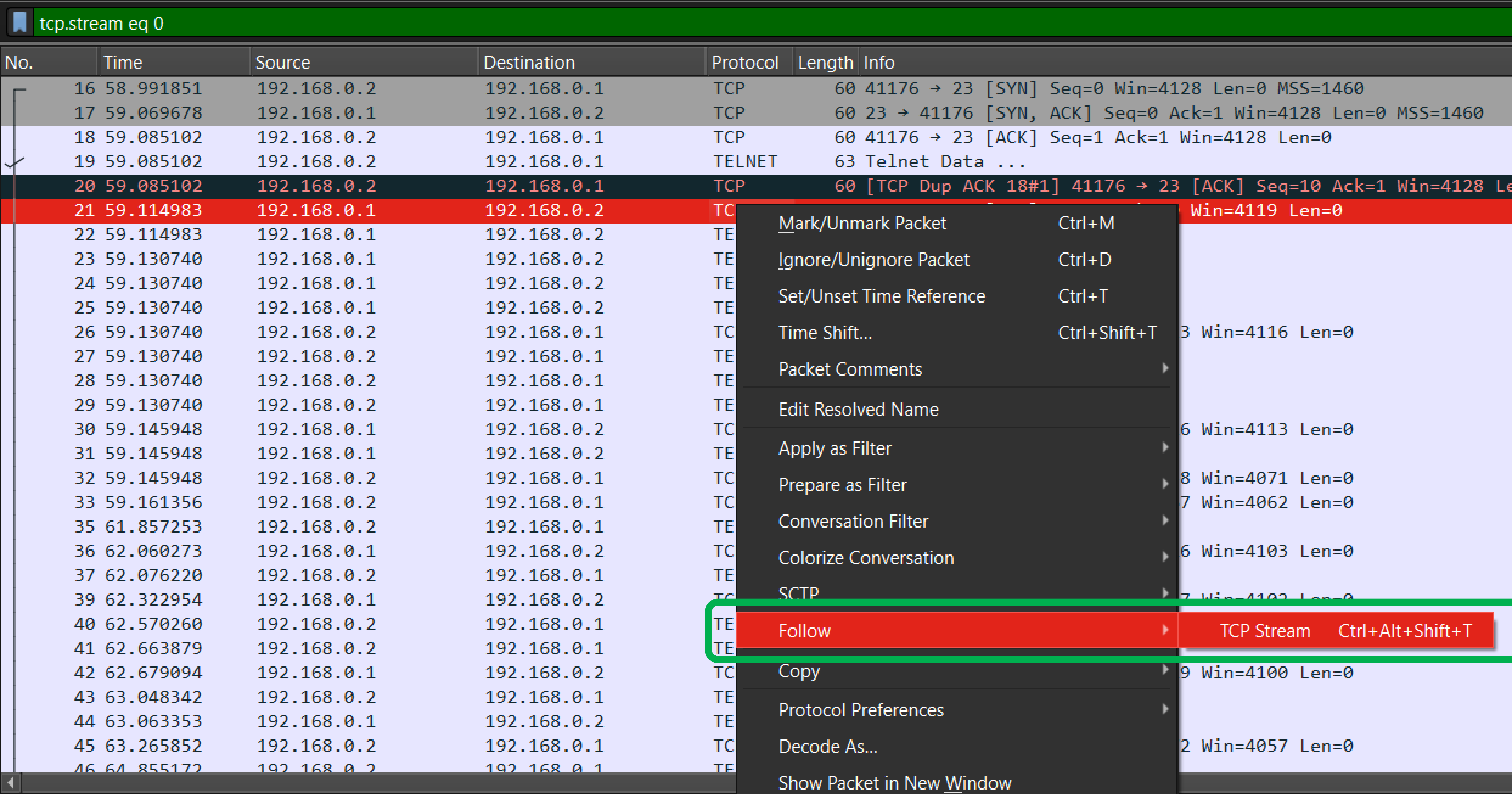
Packets: 66 · Displayed: 47 (71.2%) · Dropped: 0 (0.0%)

Profile: Default

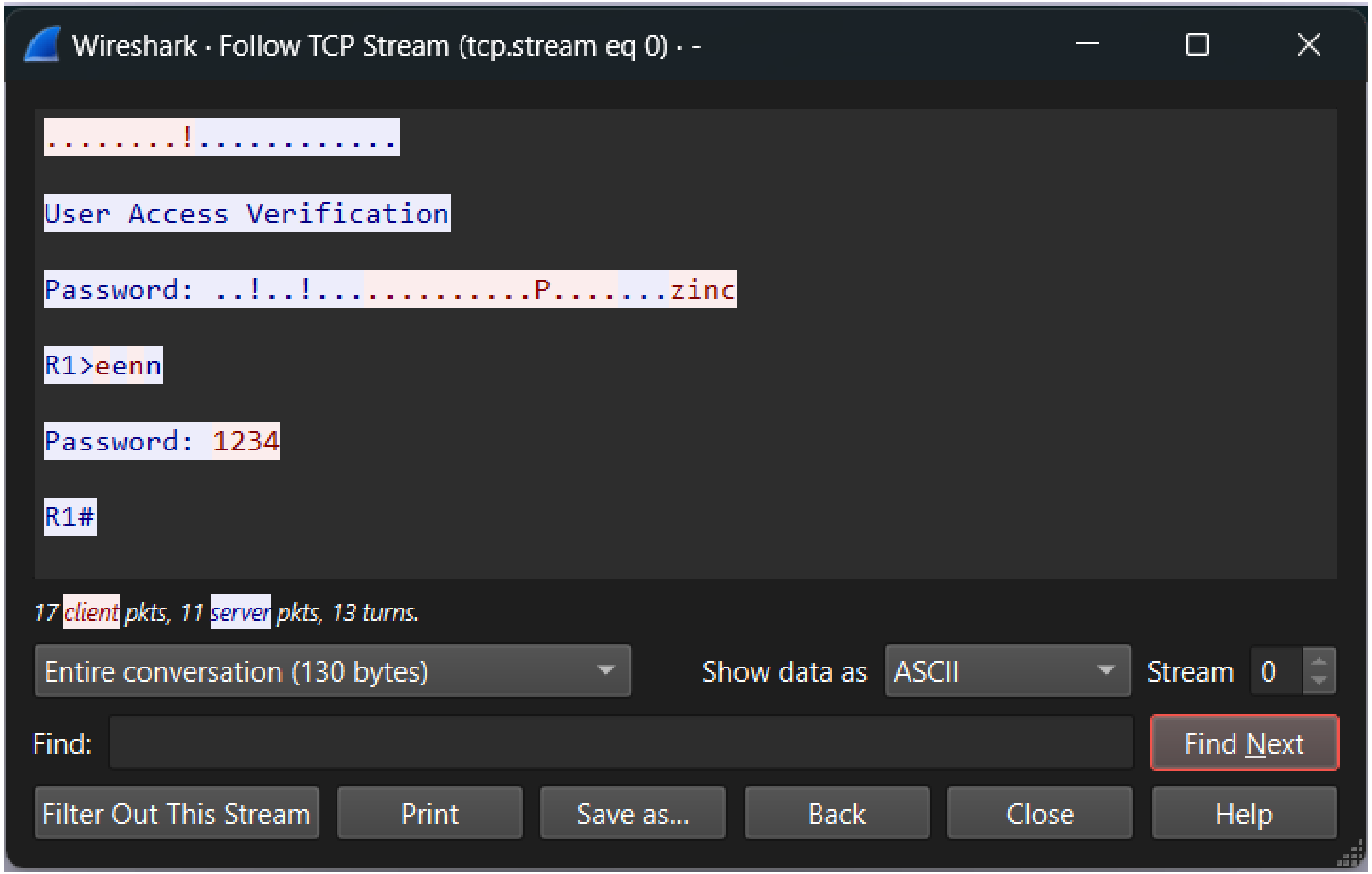
Search

ENG

7:54:53 PM
2024-07-25


tcp.stream eq 0
No. Time Source Destination Protocol Length Info
16 58.991851 192.168.0.2 192.168.0.1 TCP 60 41176 → 23 [SYN] Seq=0 Win=4128 Len=0 MSS=1460
17 59.069678 192.168.0.1 192.168.0.2 TCP 60 23 → 41176 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=1460
18 59.085102 192.168.0.2 192.168.0.1 TCP 60 41176 → 23 [ACK] Seq=1 Ack=1 Win=4128 Len=0
19 59.085102 192.168.0.2 192.168.0.1 TELNET 63 Telnet Data ...
20 59.085102 192.168.0.2 192.168.0.1 TCP 60 [TCP Dup ACK 18#1] 41176 → 23 [ACK] Seq=10 Ack=1 Win=4128 Le
21 59.114983 192.168.0.1 192.168.0.2 Win=4119 Len=0
22 59.114983 192.168.0.1 192.168.0.2
23 59.130740 192.168.0.1 192.168.0.2
24 59.130740 192.168.0.1 192.168.0.2
25 59.130740 192.168.0.1 192.168.0.2
26 59.130740 192.168.0.2 192.168.0.1 Win=4116 Len=0
27 59.130740 192.168.0.2 192.168.0.1
28 59.130740 192.168.0.2 192.168.0.1
29 59.130740 192.168.0.2 192.168.0.1
30 59.145948 192.168.0.1 192.168.0.2 Win=4113 Len=0
31 59.145948 192.168.0.1 192.168.0.2
32 59.145948 192.168.0.2 192.168.0.1 Win=4071 Len=0
33 59.161356 192.168.0.2 192.168.0.1 Win=4062 Len=0
35 61.857253 192.168.0.2 192.168.0.1
36 62.060273 192.168.0.1 192.168.0.2 Win=4103 Len=0
37 62.076220 192.168.0.2 192.168.0.1
39 62.322954 192.168.0.1 192.168.0.2
40 62.570260 192.168.0.2 192.168.0.1
41 62.663879 192.168.0.2 192.168.0.1
42 62.679094 192.168.0.1 192.168.0.2 Win=4100 Len=0
43 63.048342 192.168.0.2 192.168.0.1
44 63.063353 192.168.0.1 192.168.0.2
45 63.265852 192.168.0.2 192.168.0.1 Win=4057 Len=0
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
TCP Stream Ctrl+Alt+Shift+T
Copy
Protocol Preferences
Decode As...
Show Packet in New Window

Ohhh No... ➔ Hacker See every things



Private & Public IP Address

Q: why IPv4 still used?

Private & Public IP Address technical

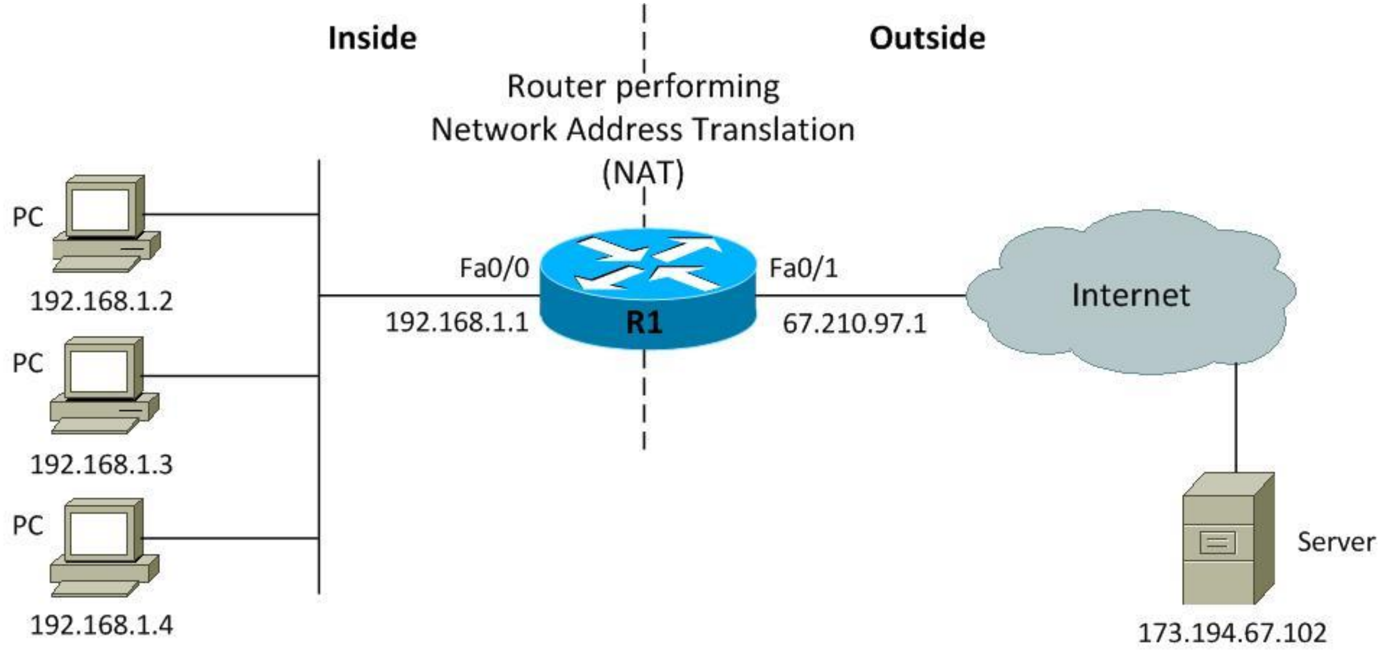
- Private IP addresses are used within a private network and are not routable on the internet. They are intended for internal use within an organization or home. **(Free)**

- Public IP addresses are assigned by Internet Service Providers (ISPs) and are routable on the internet. They are unique and must be registered with the Internet Assigned Numbers Authority (IANA). **(sale $$)**

Private IP address space	
From	**To**
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

NAT (Network Address Translation)

- Routers use NAT to map private IP addresses to a public IP address, allowing multiple devices on a private network to share a single public IP address when accessing the internet.

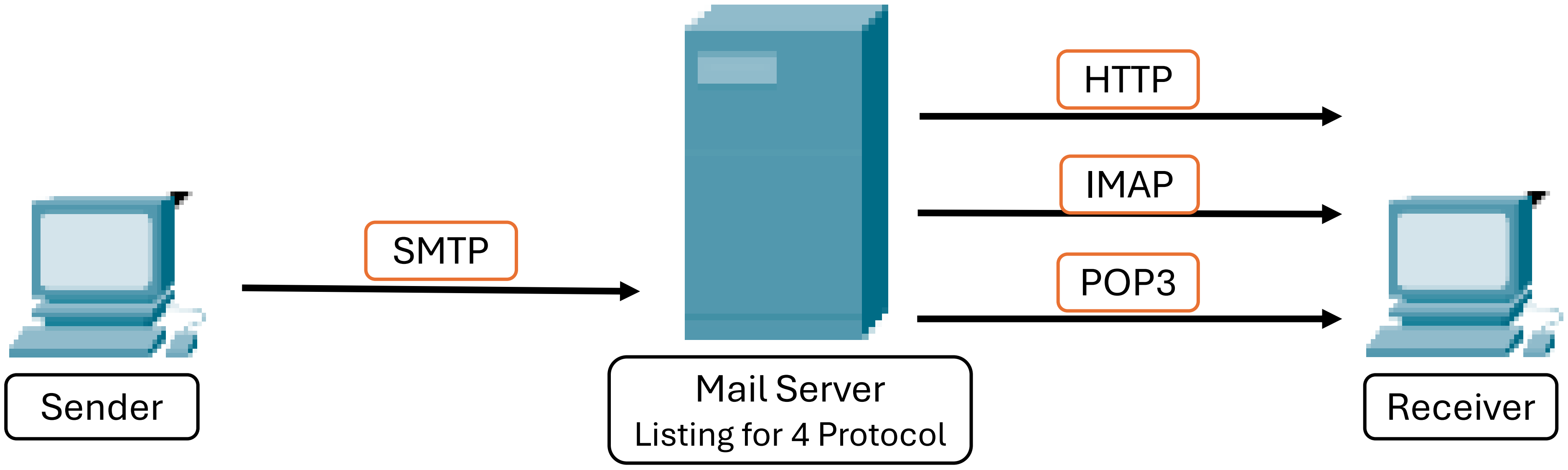


NAT Translation Table

Protocol	Inside Local IP : Port	Inside Global IP : Port
ICMP	192.168.1.2 : 18	67.210.97.1 : 18
ICMP	192.168.1.3 : 19	67.210.97.1 : 19
ICMP	192.168.1.4 : 20	67.210.97.1 : 20

Mail Protocols

- HW: what difference between SMTP, HTTP, IMAP, and POP3?
- HW: FTP use two ports 20 and 21,what are the differences between them?!
- HW: what is TFTP Protocol?

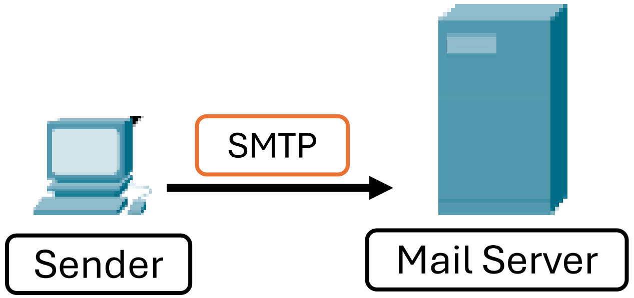


Ans:

1. **SMTP (Simple Mail Transfer Protocol)**:

Used for sending emails, uses port 25, can also use ports 587 or 465 for encrypted transmission.

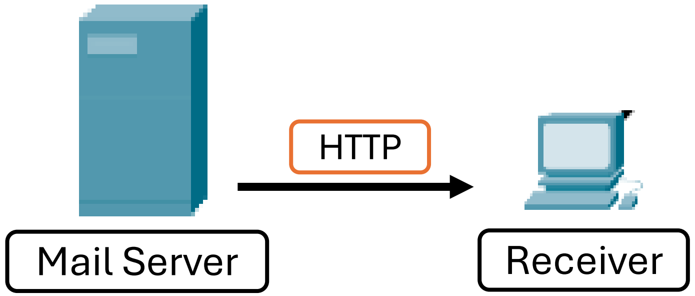
Facilitates the sending of email messages between servers. It's used by mail servers to relay outgoing mail.



2. **HTTP (Hypertext Transfer Protocol):**

Used for transferring hypertext documents, uses port 80 standard 443 for HTTPS secure.

Defines how messages are formatted and transmitted, and how web servers and browsers should respond to various commands. It's the foundation of any data exchange on the Web.

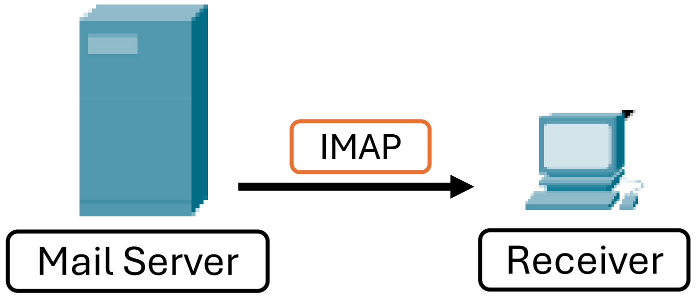


Cont...

3. **IMAP (Internet Message Access Protocol):**

Used for retrieving and storing email on a mail server, port 143 for standard, port 993 for encrypted.

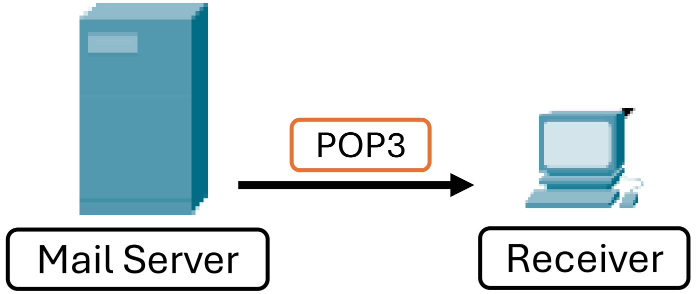
Allows multiple clients to manage and access the same mailbox, providing the ability to view and organize emails on the server.



4. **POP3 (Post Office Protocol version 3):**

Used for retrieving email from a mail server, port 110 for standard and 995 for encrypted.

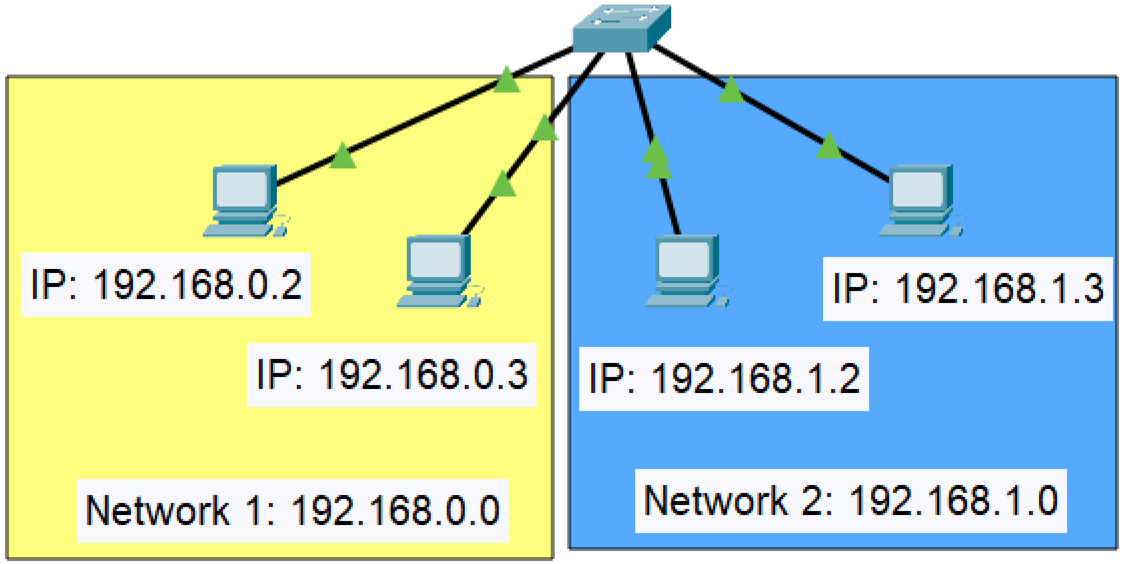
Downloads emails from the server to the client and, typically, deletes them from the server afterward. It's designed for users to download their email and then disconnect from the mail server.



Subnetting

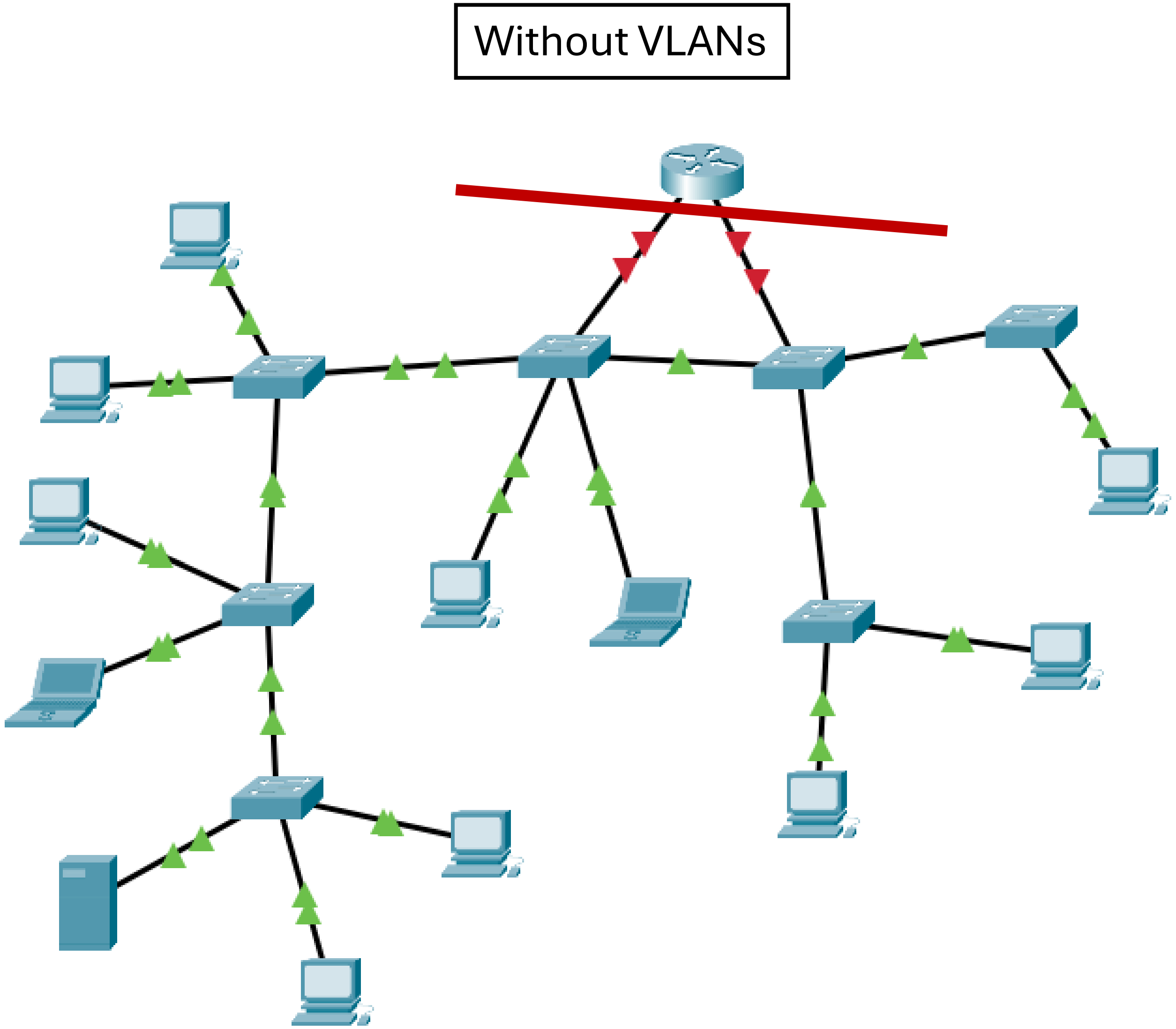
- Problem:
- Subnetting & Subnet mask & IP Address in PC Level…
- PC know its network.
- PC can change its IP Address ➔ sends message to other networks (less secure)

- Q: BC message sends to one or different networks?
- ARP:
- DHCP:
- Relay DHCP:



Broadcast Domain

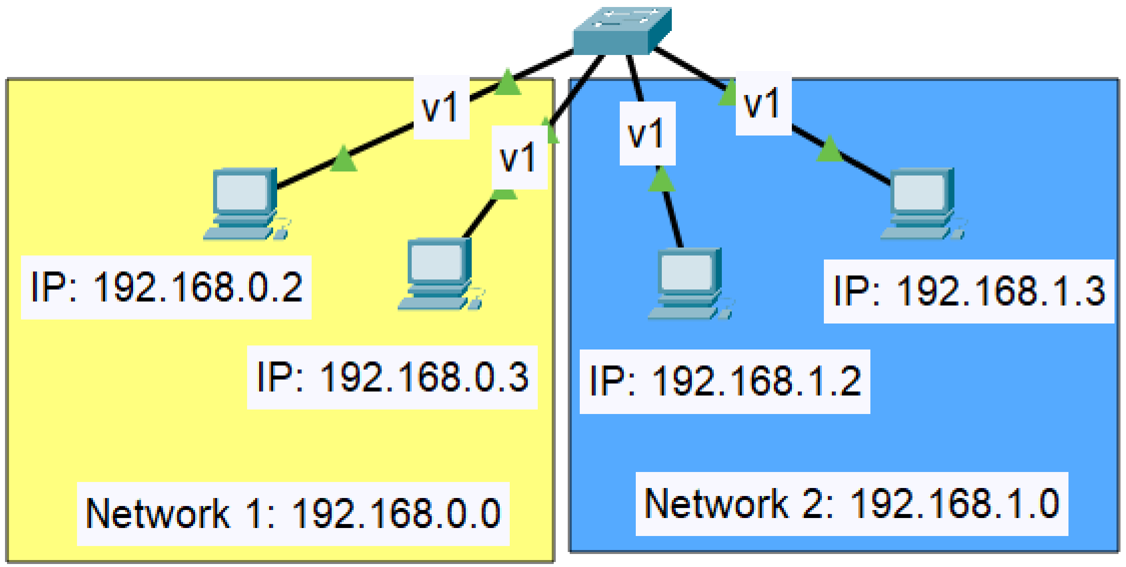
- Same network ➔ End at first router or VLAN.
- So VLANs minimize traffic of broadcast.

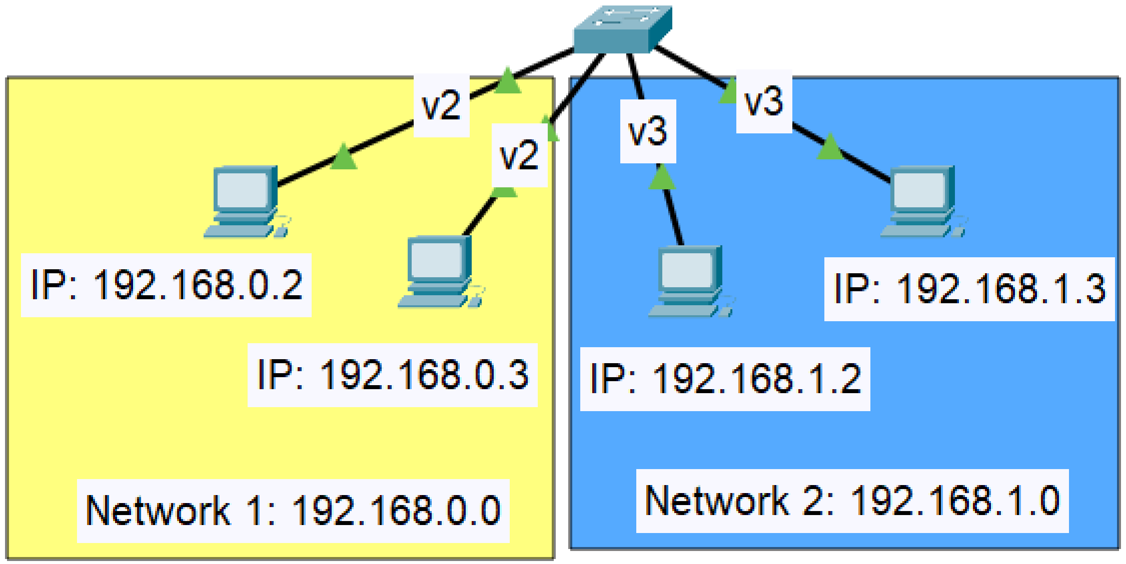


VLAN (L2 Switch)

- VLANs (Virtual Local Area Networks) are used to segment the network into different broadcast domains. This helps improve security and performance by isolating network traffic.

- At the switching level in the data link layer, it is configured in the switch device.
- Users cannot see or change it, and they do not know which VLAN it is in and how many VLANs are in the topology.
- If a user changes his IP address, he will not connect to other VLANs.
- By default, all devices connected to the switch are in VLAN 1 (the default VLAN), and they are all in the same VLAN.





Day 5

- Outline
  - Subnetting & VLANs
    - Broadcast in VLANs
    - 2 VLANs Project
    - Switch Ports Modes
      - Access Mode
      - Trunk
      - Dynamic:Auto
      - dynamicDesirable
    - Intra VLANs
    - Router sub-interface & .1q protocol
  - Wires
  - Project in IntrVLAN
  - Routing
  - Static VS Dynamic Routing
  - Router Simple Project

Switch Show VLANs

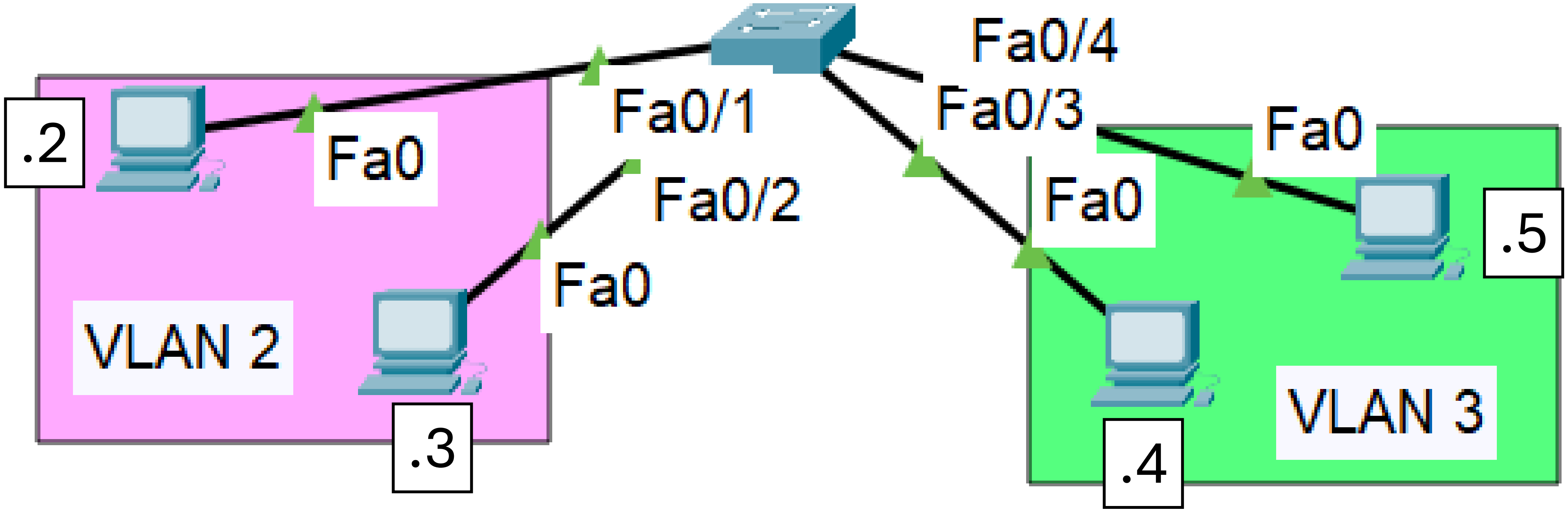
Switch>enable

Switch#**show 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	

Example 1

All same Network: 192.168.0



Example 1

Switch#configure terminal

Switch(config)#**vlan** 2 — Create VLAN 2

Switch(config-vlan)#exit

Switch(config)#**vlan** 3 — Create VLAN 3

Switch(config-vlan)#exit

Switch(config)#**int**erface **f**astEthernet**0/1**

Switch(config-if)#**sw**itchport **m**ode **a**ccess

Switch(config-if)#**sw**itchport **a**ccess **v**lan 2

Put VLAN 2 in port f0/1 & select mode

Switch(config-if)#do write

Switch(config-if)#exit

Switch(config)#**int**erface **f**astEthernet**0/2**

Switch(config-if)#**sw**itchport **m**ode **a**ccess

Switch(config-if)#**sw**itchport **a**ccess **v**lan 2

Put VLAN 2 in port f0/2 & select mode

Switch(config-if)#do write

Example 1

Switch(config)#**int**erface **f**astEthernet**0/3**

Switch(config-if)#**sw**itchport **m**ode **a**ccess

Switch(config-if)#**sw**itchport **a**ccess **v**lan 3

Put VLAN 3 in port f0/3 & select mode

Switch(config-if)#do write

Switch(config-if)#exit

Switch(config)#**int**erface **f**astEthernet**0/4**

Switch(config-if)#**sw**itchport **m**ode **a**ccess

Switch(config-if)#**sw**itchport **a**ccess **v**lan 3

Put VLAN 3 in port f0/4 & select mode

Switch(config-if)#do write

Switch Show VLANs

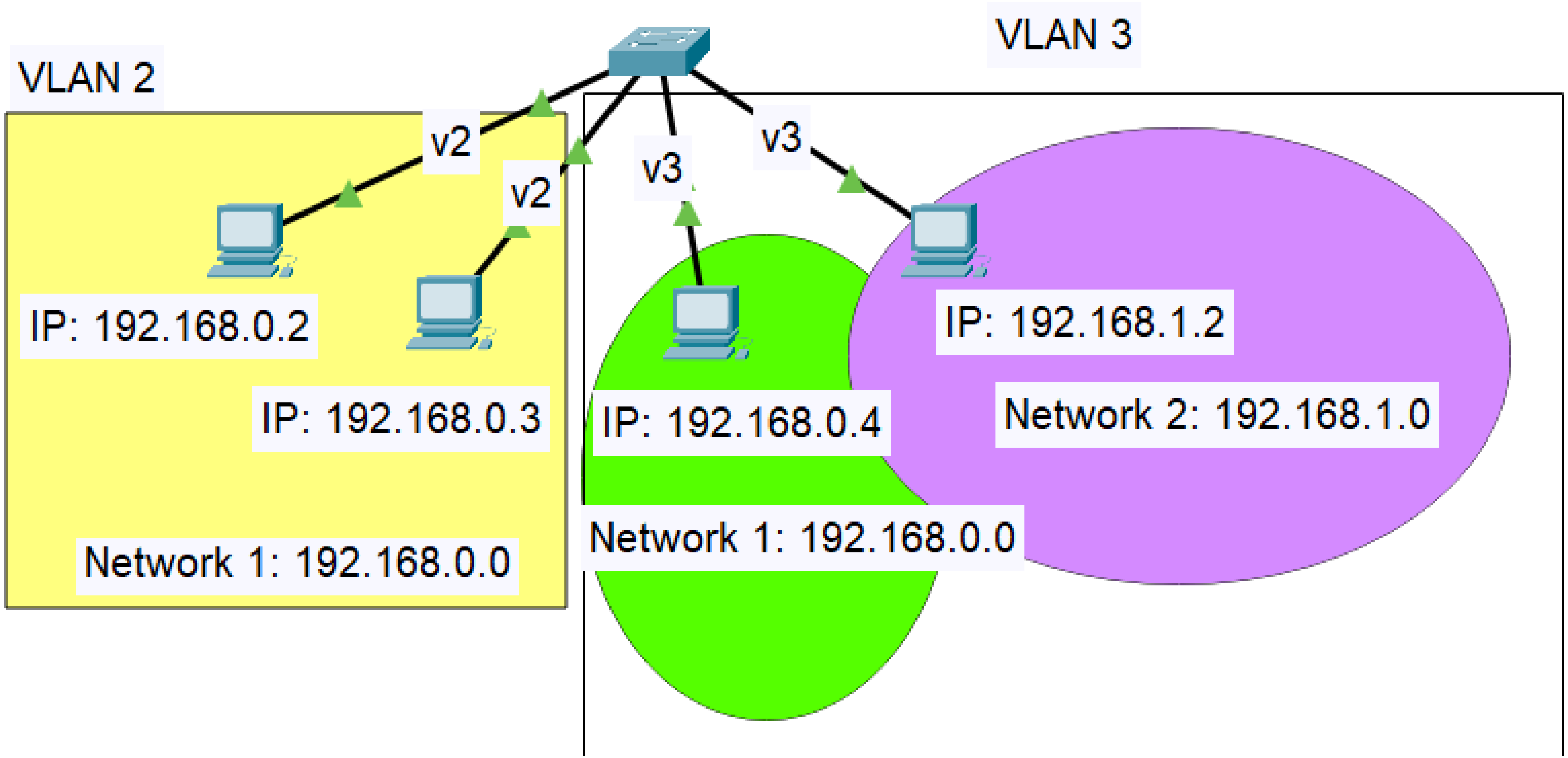
Switch#**show vlan brief**

VLAN	Name	Status	Ports
----	--------------------------------	---------	-------------------------------
1	default	active	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
2	**VLAN0002**	**active**	**Fa0/1, Fa0/2**
3	**VLAN0003**	**active**	**Fa0/3, Fa0/4**
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Connection

Network	VLAN	Connection
Same	Same	✓
Same	Different	✗
Different	Same	✗
Different	Different	✗

Example 2



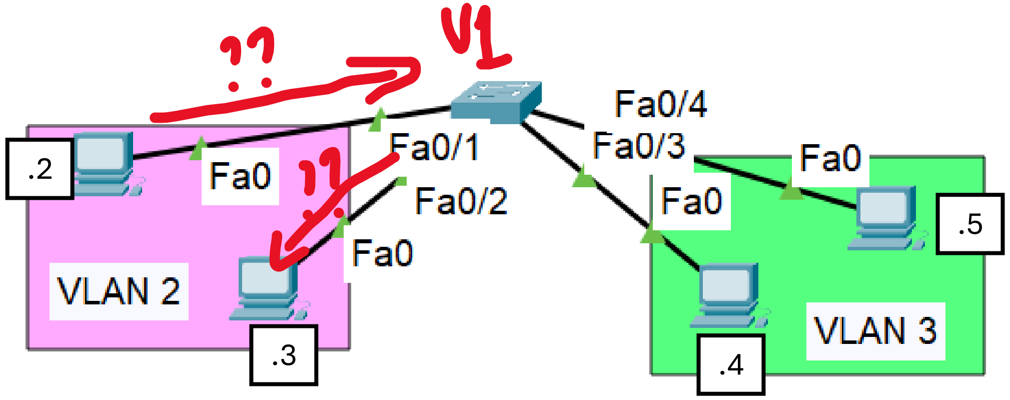
VLAN 2
VLAN 3
v2
v2
v3
v3
IP: 192.168.0.2
IP: 192.168.0.3
IP: 192.168.0.4
IP: 192.168.1.2
Network 2: 192.168.1.0
Network 1: 192.168.0.0
Network 1: 192.168.0.0

Switch Port Models

Switch ports can be configured in different modes:

1. Access Mode:

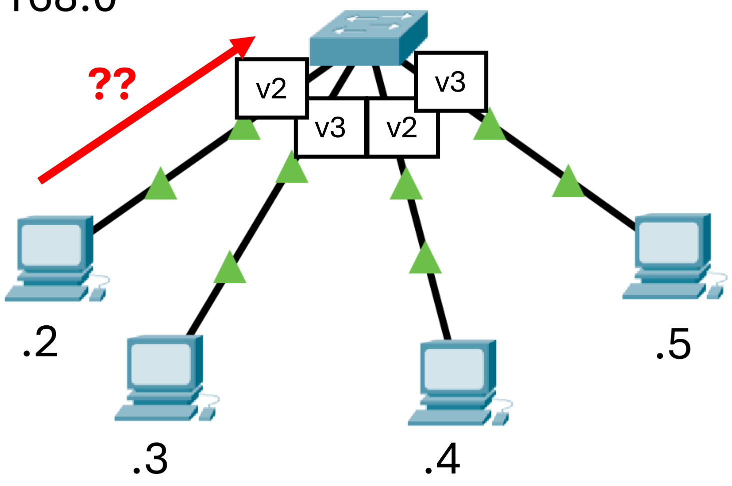
is a port configuration mode used on switches for connecting end devices such as computers, printers, and servers.

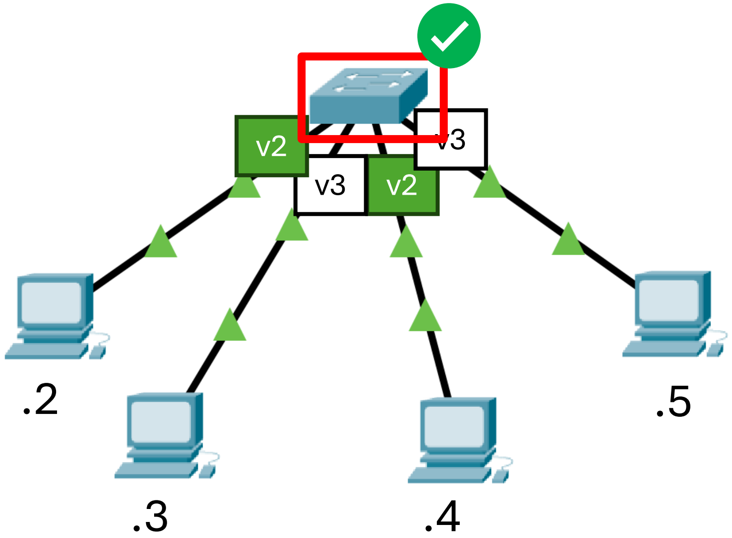


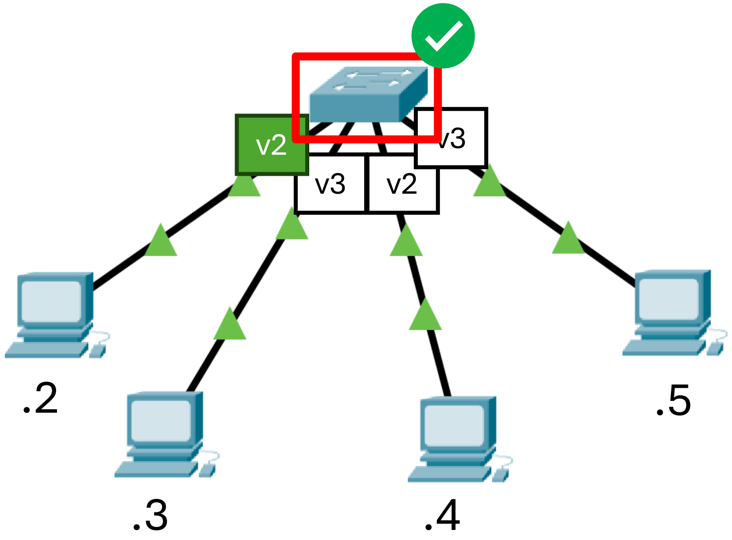
- The port is assigned to a **single VLAN.**
- All traffic coming in and out of the port is **untagged,** meaning that VLAN information is not included in the Ethernet frames.
- Devices connected to an access port are **unaware of VLANs.**

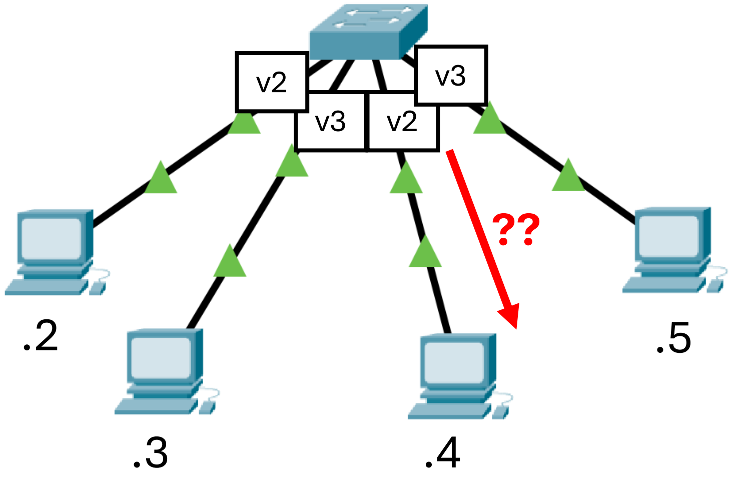
Access Mode Used

NID: 192.168.0
SM: /24

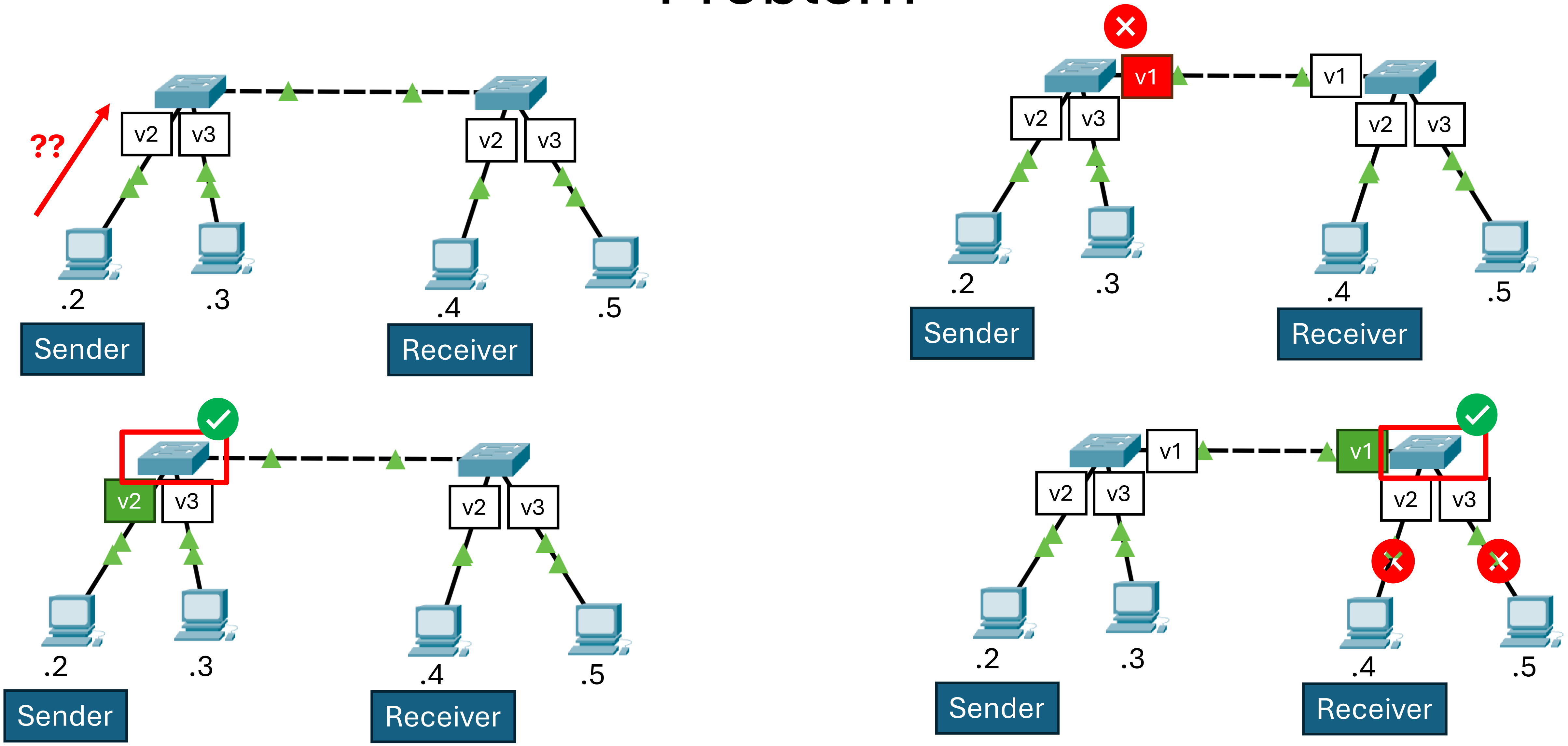








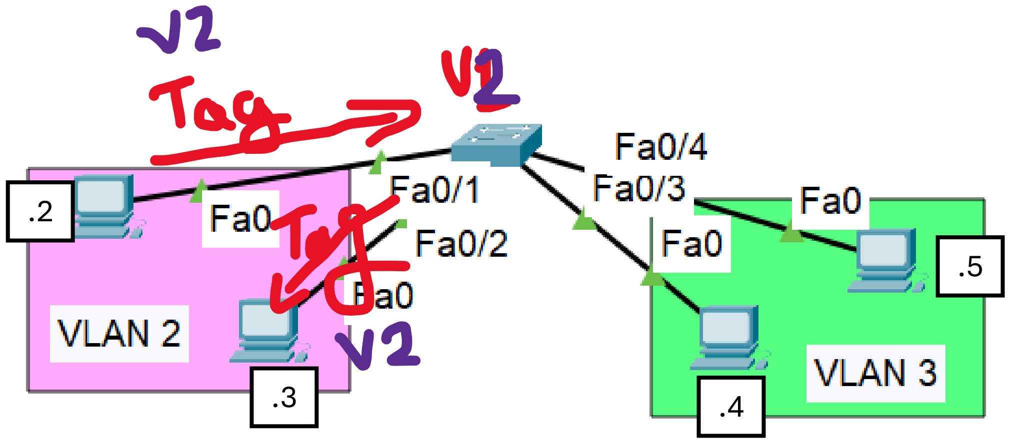
Problem



??
v2
v3
v2
v3
.2
.3
.4
.5
Sender
Receiver
v1
v2
v3
v1
v2
v3
.2
.3
.4
.5
Sender
Receiver
v2
v3
v2
v3
.2
.3
.4
.5
Sender
Receiver
v1
v2
v3
v1
v2
v3
.2
.3
.4
.5
Sender
Receiver

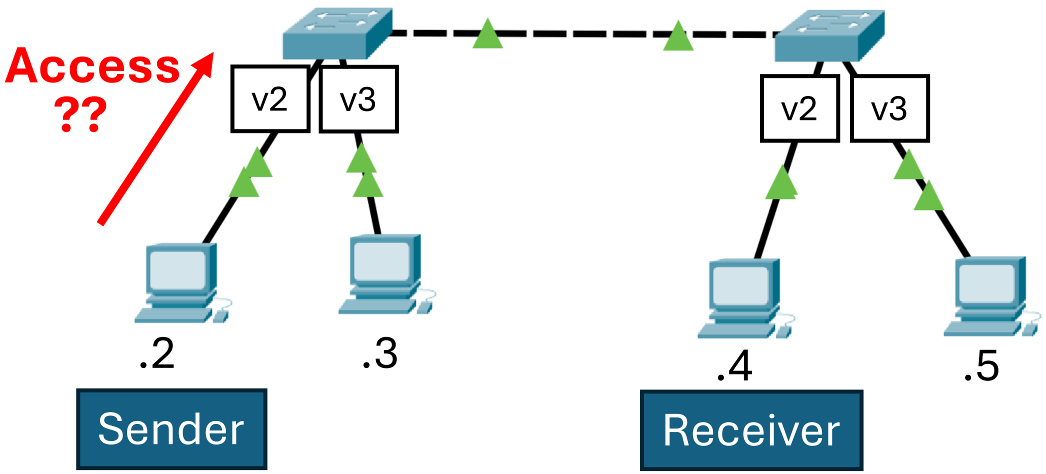
2. Trunk Mode

- is used on switch ports to carry traffic for multiple VLANs used **between routers and servers**.
- can carry traffic from **multiple VLANs**.
- Traffic on a trunk port is **tagged** with VLAN identifiers (**DTP**).
- A trunk port can have a **native VLAN**, which is the VLAN for untagged traffic. By default, the native VLAN is VLAN 1.

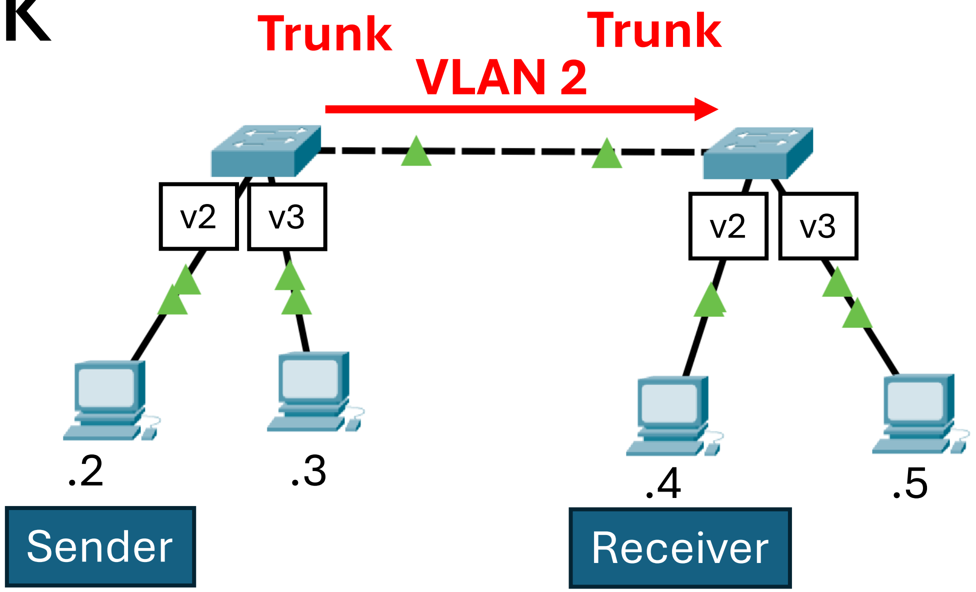


- Q: Native VLAN?

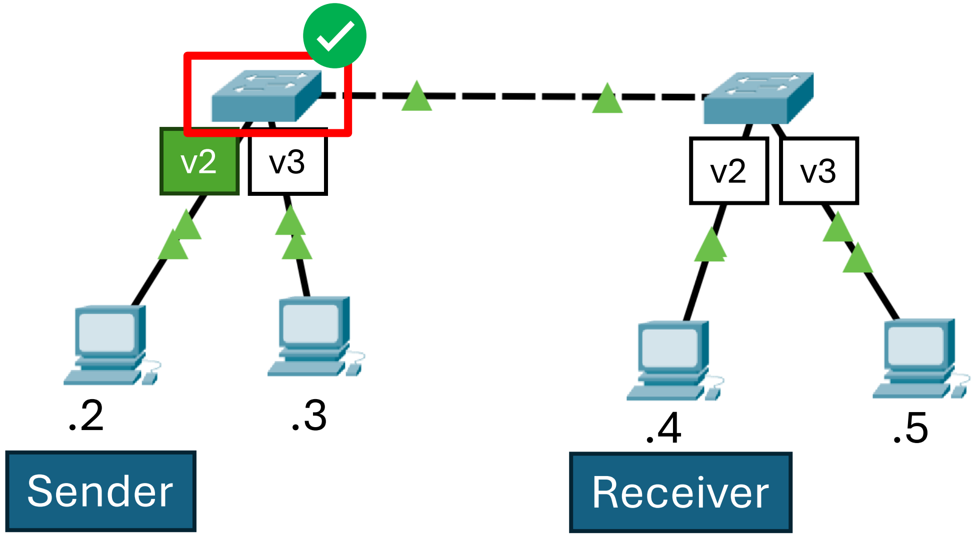
Use Trunk



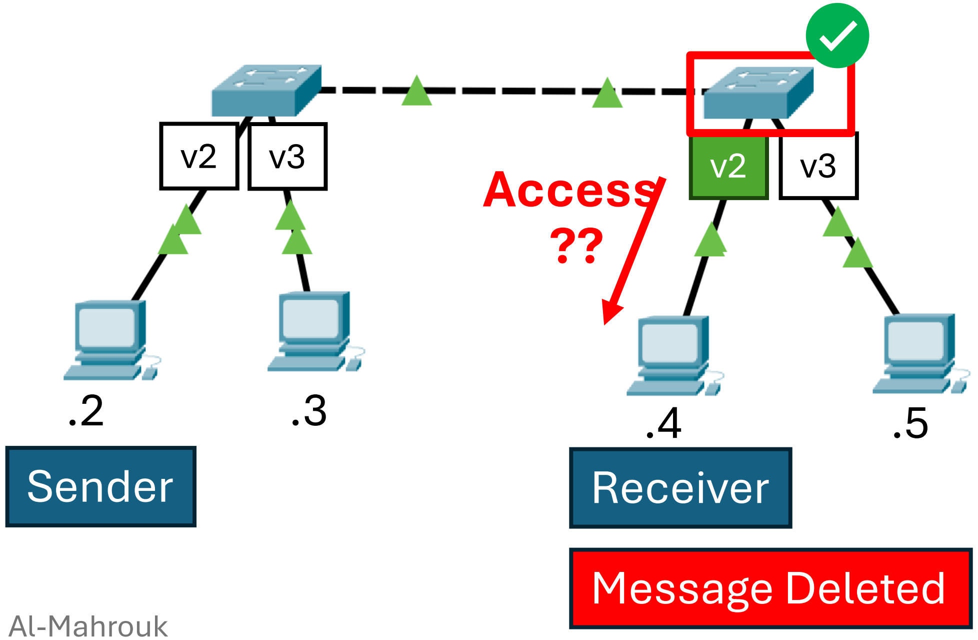
Access
??
v2
v3
v2
v3
.2
.3
.4
.5
Sender
Receiver



Trunk
Trunk
VLAN 2
v2
v3
v2
v3
.2
.3
.4
.5
Sender
Receiver



v2
v3
v2
v3
.2
.3
.4
.5
Sender
Receiver



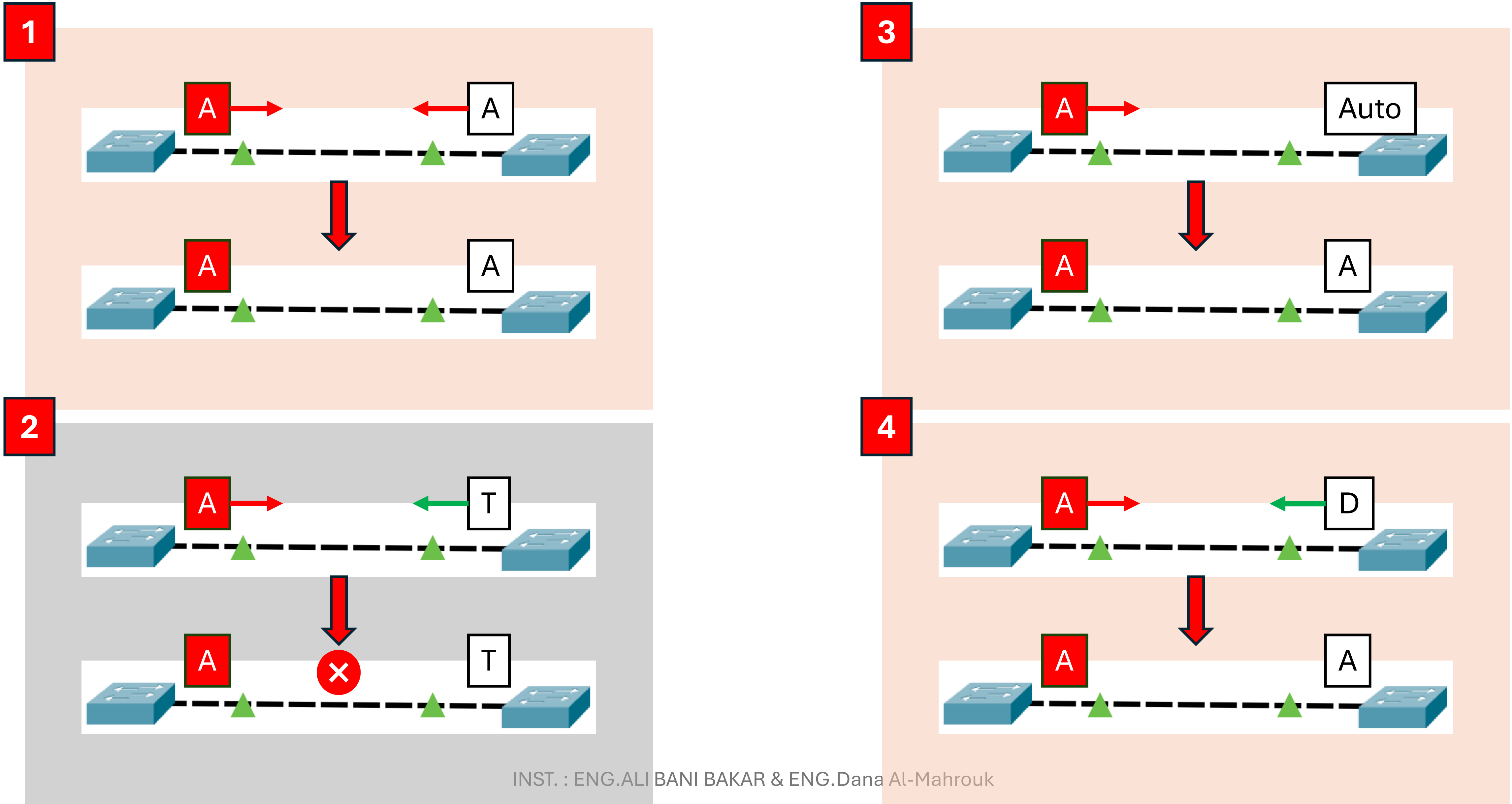
v2
v3
Access
??
v2
v3
.2
.3
.4
.5
Sender
Receiver
Message Deleted

Dynamic Mode (Auto and Desirable Modes)

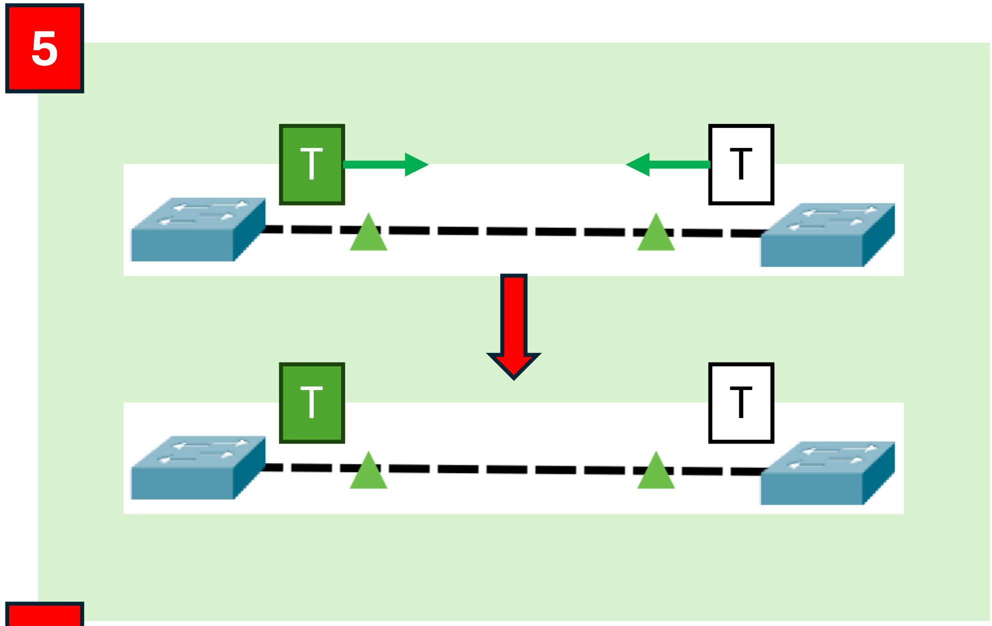
- DTP is a Cisco proprietary protocol that negotiates trunking on a link between two devices and helps manage the trunk links dynamically.

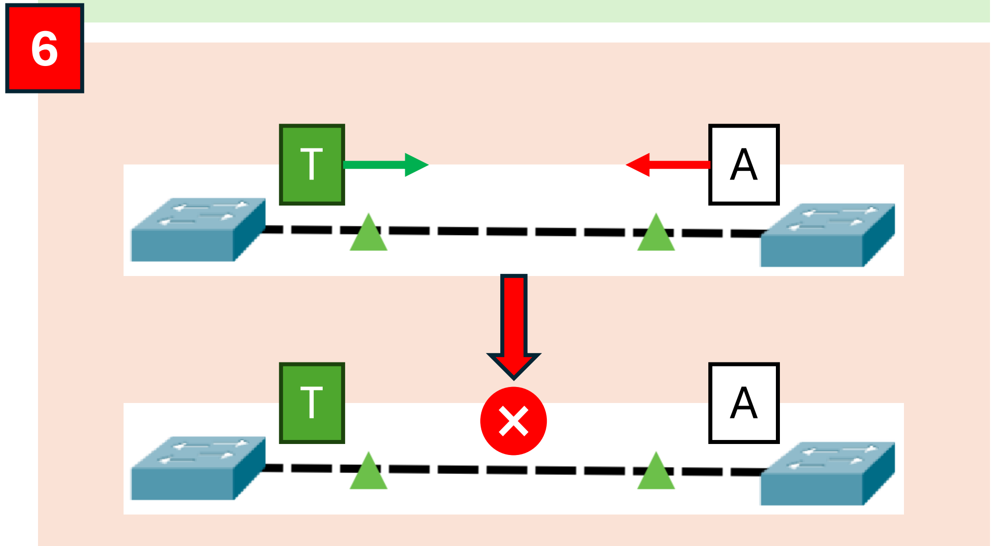
	Access	Trunk	Auto	Desirable
Access	A	❌	A	A
Trunk	❌	T	T	T
Auto	A	T	A	T
Desirable	A	T	T	T

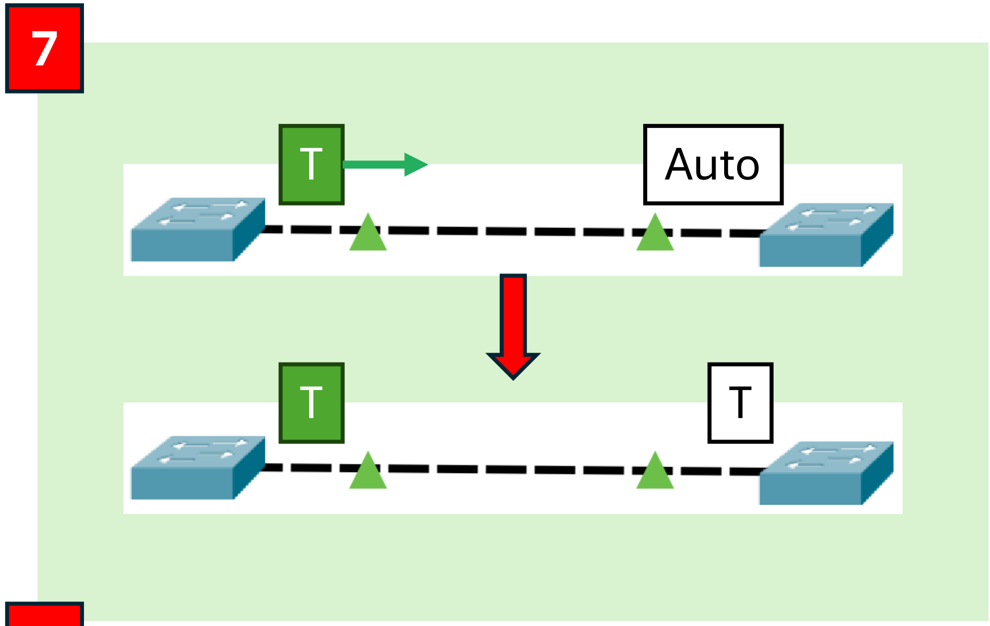
Dynamic Trunking Protocol (1. Access)

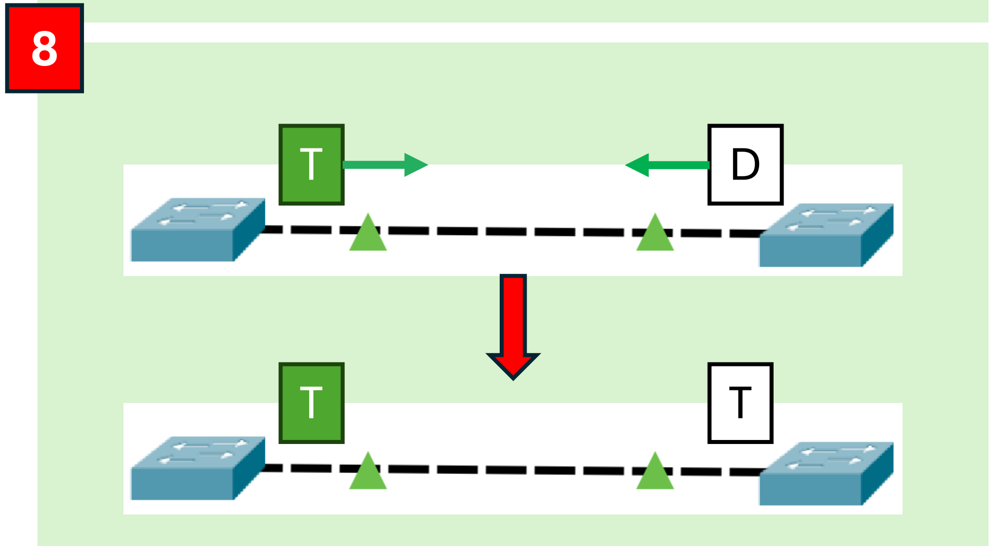


Dynamic Trunking Protocol (2. Trunk)

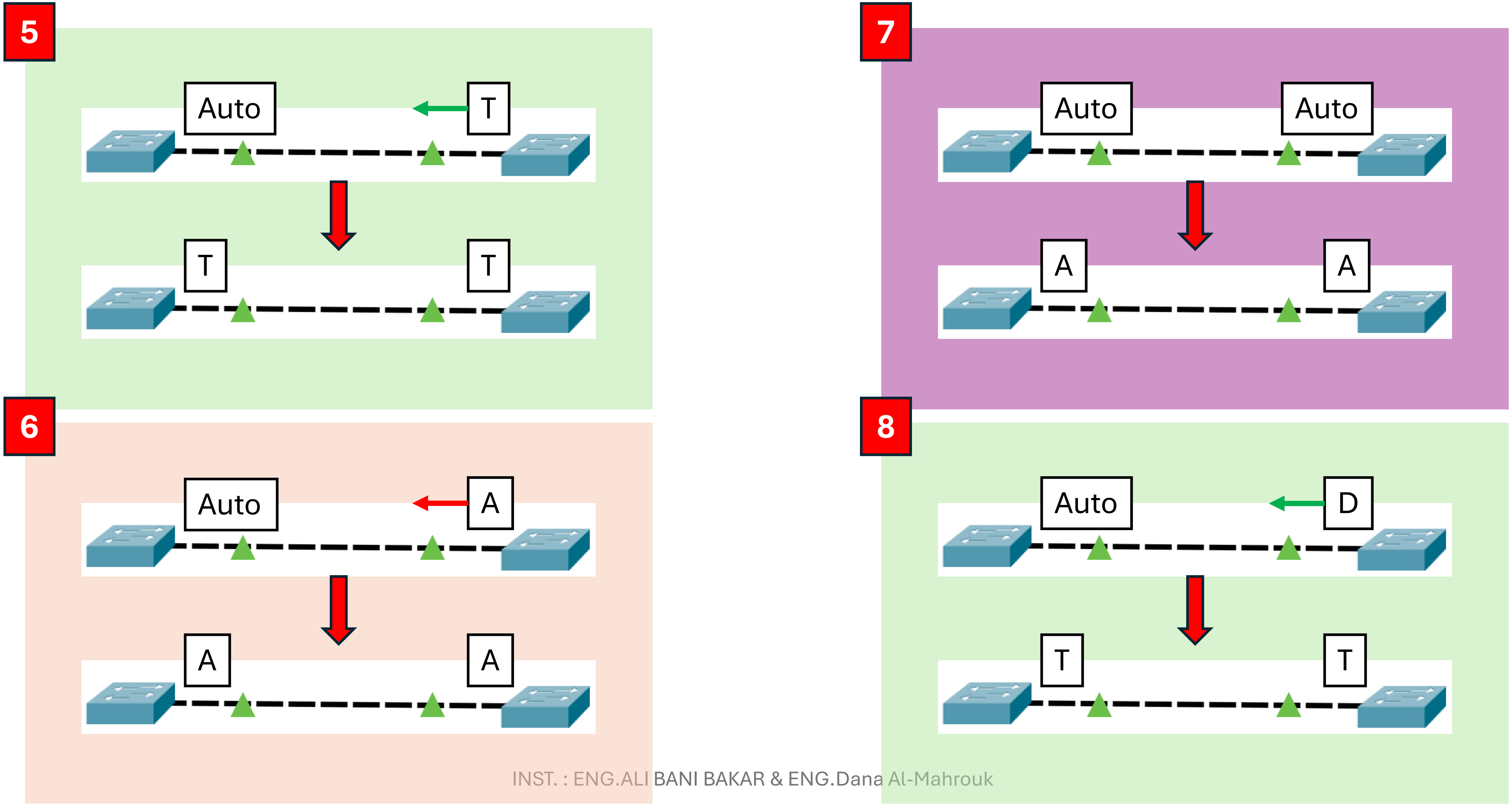






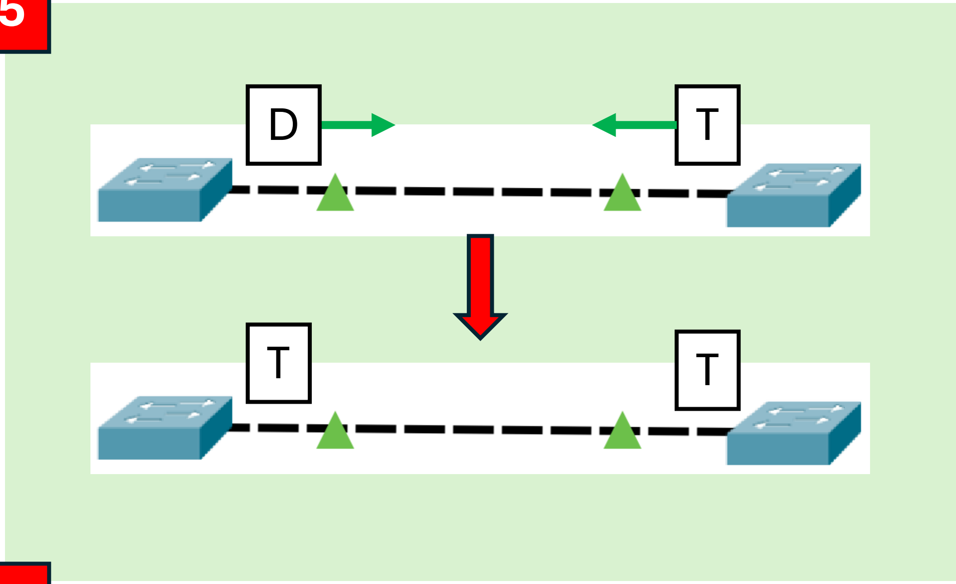


Dynamic Trunking Protocol (3. Auto)

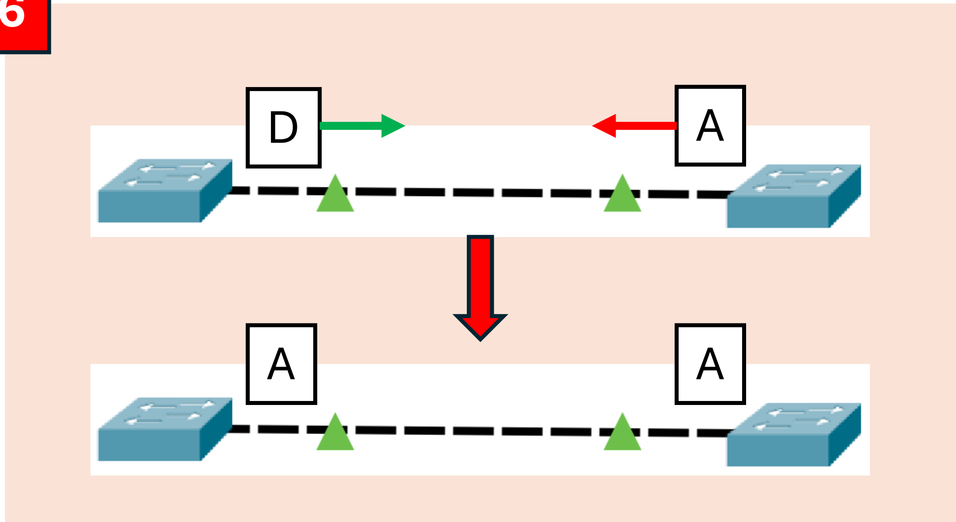


Dynamic Trunking Protocol (4. Desirable)

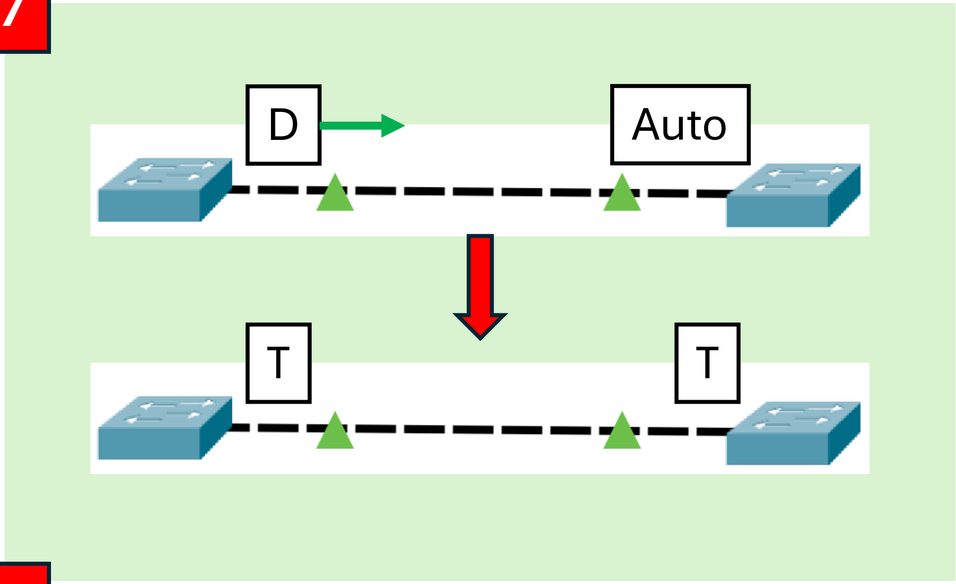
5



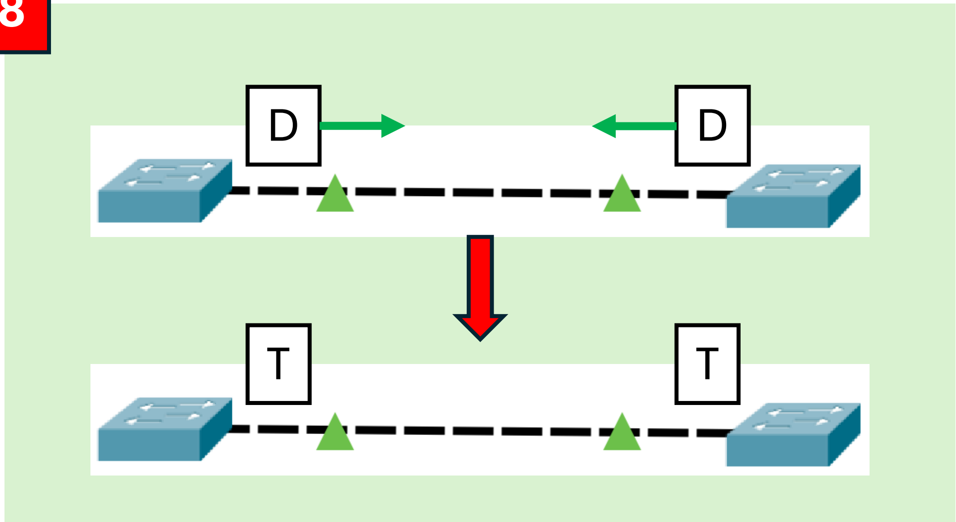
6



7

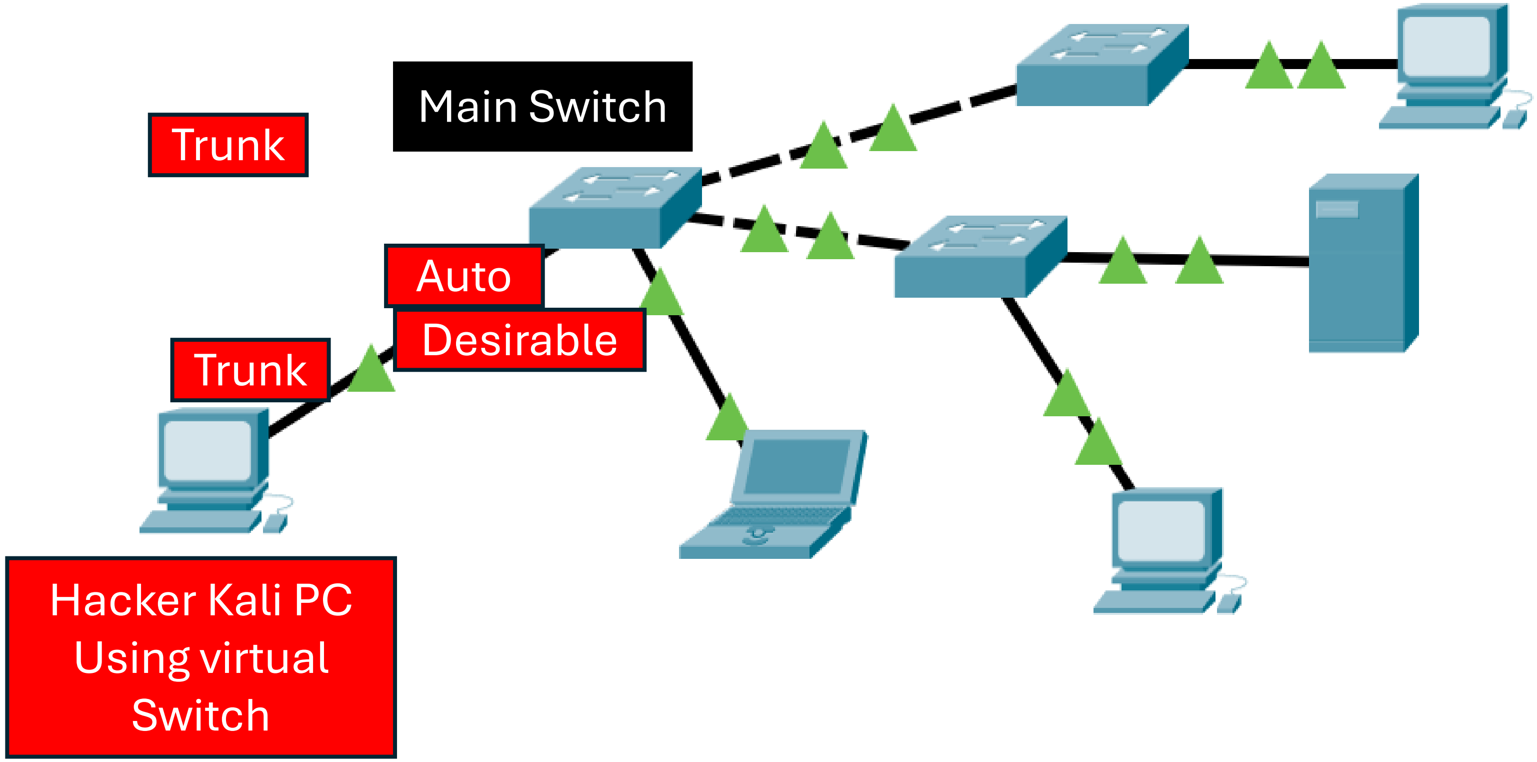


8



Hacker

Can Access to all network if main Switch in at Truck or Auto or Desirable.



Problem

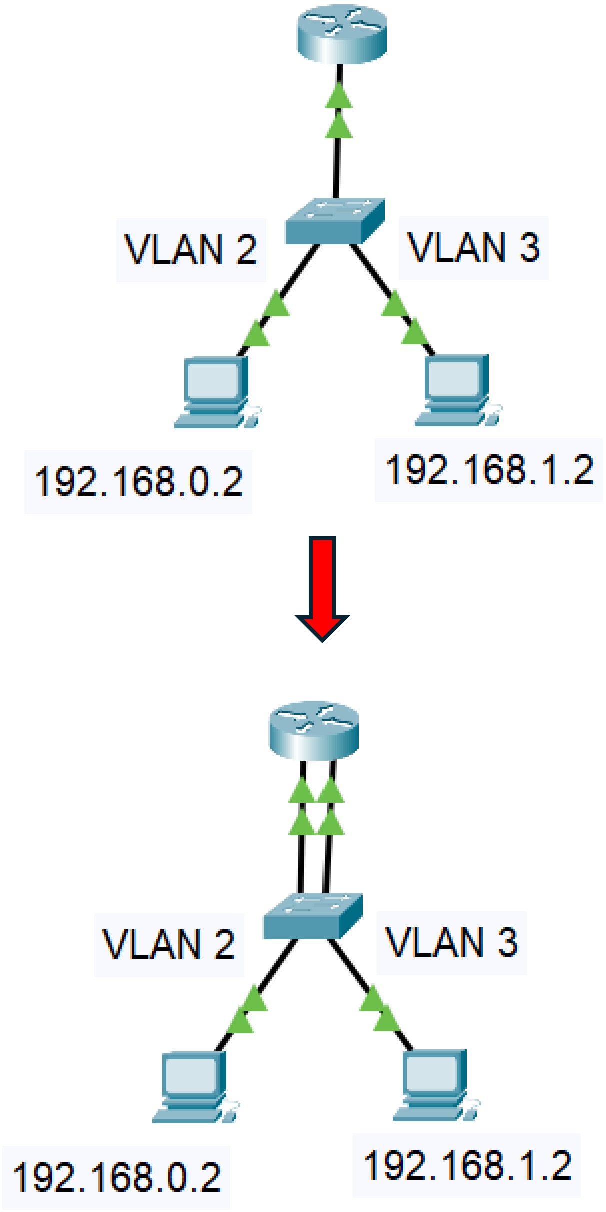
- If we need to connected between to VLANs, we need to use Router.

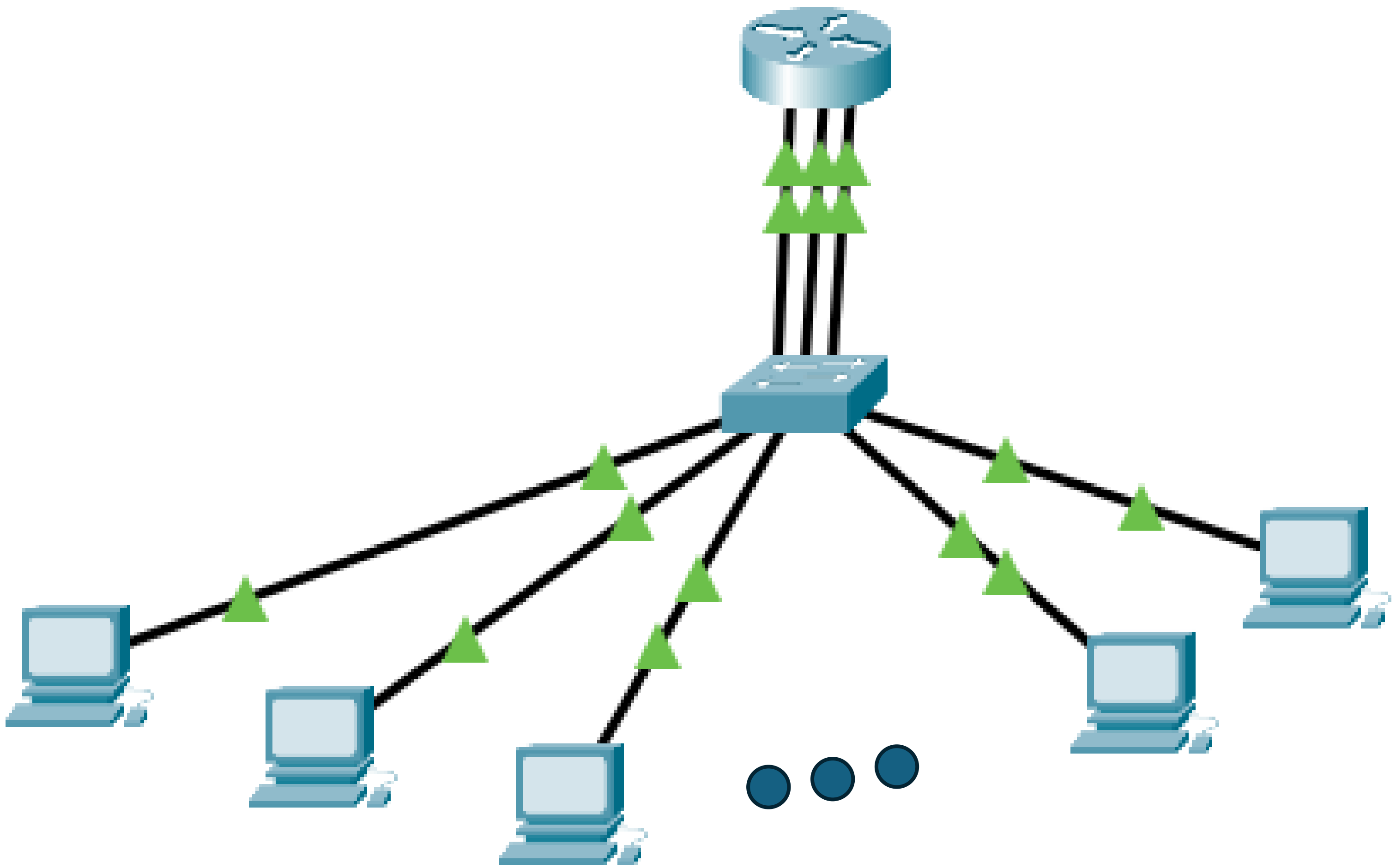
- Problems:

❖Router can not work with Trunk Mode (Header of message will change so router will delete the message, he think it wrong message because the CRC | FSC flag will change in Trunk Mode ➔ Because we add the VLAN Tag).

❖For each VLAN we need to add a new Interface in the router (so expensive).

❖When message get out of router it will not get a VLAN Tag, because router do not work with Trunk Mode.



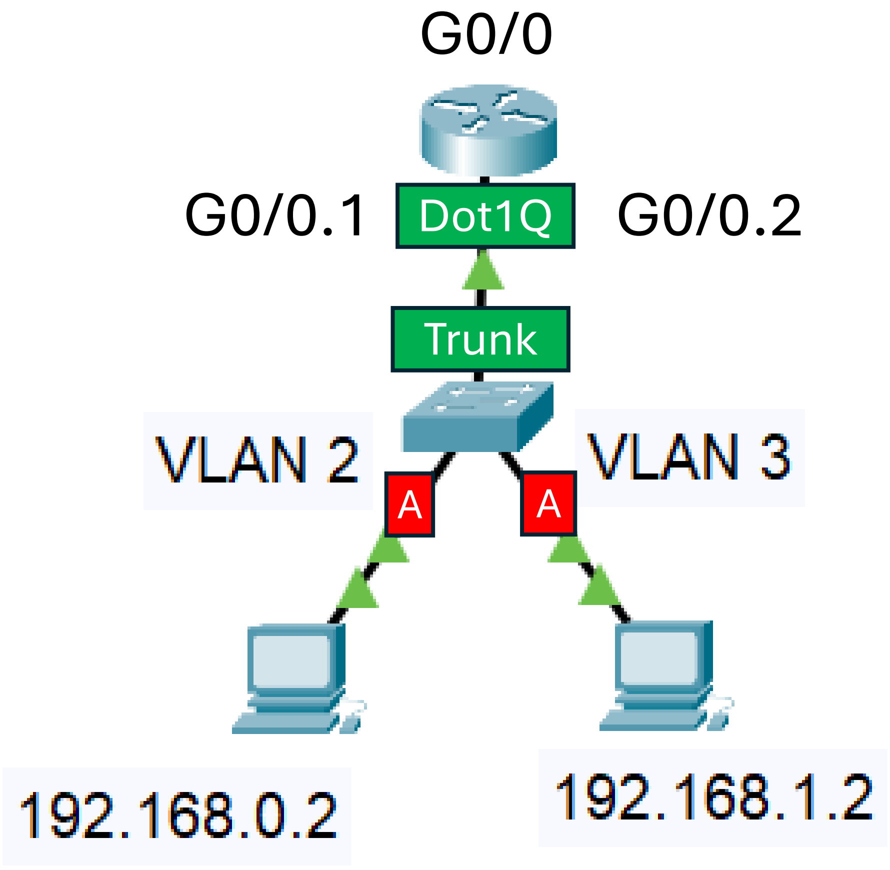


Inter VLAN

- allows communication between different VLANs.
- We need to use it with **Sub-Interface:**

A sub-interface in a router is a logical interface created within a physical interface.

Allowing multiple VLANs to share a single physical interface while remaining logically separated.



Dot1Q:

protocol is a networking standard that supports VLANs (Virtual Local Area Networks) on an IEEE 802.3 (Ethernet) network.

When a frame enters a VLAN-aware switch or router, the device adds an 802.1Q tag to the frame header if it is to be sent out on a trunk link.

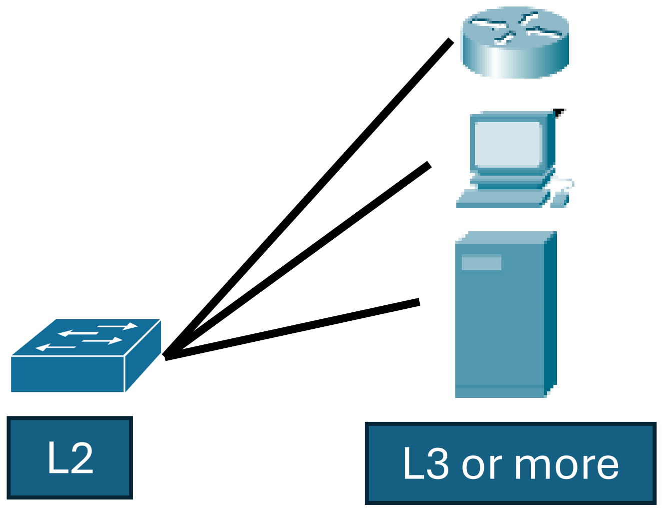
Wire

• **Straight**-through Ethernet Cable:

used to connect different types of devices. It has the same wiring standard on both ends.

Uses:

- Connecting a computer to a switch or hub
- Connecting a router to a modem
- Connecting different network devices to a central networking device

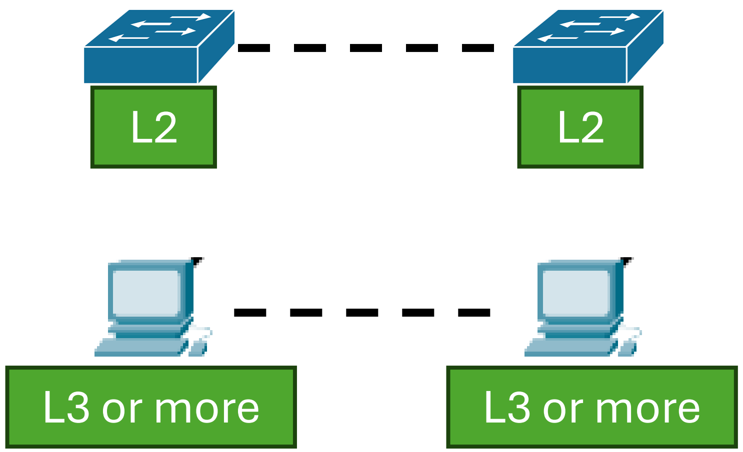


• **Crossover** Ethernet Cable:

used to connect similar types of devices directly, without the need for a switch or hub. The internal wiring swaps the transmit and receive signal pairs, allowing two devices to communicate directly.

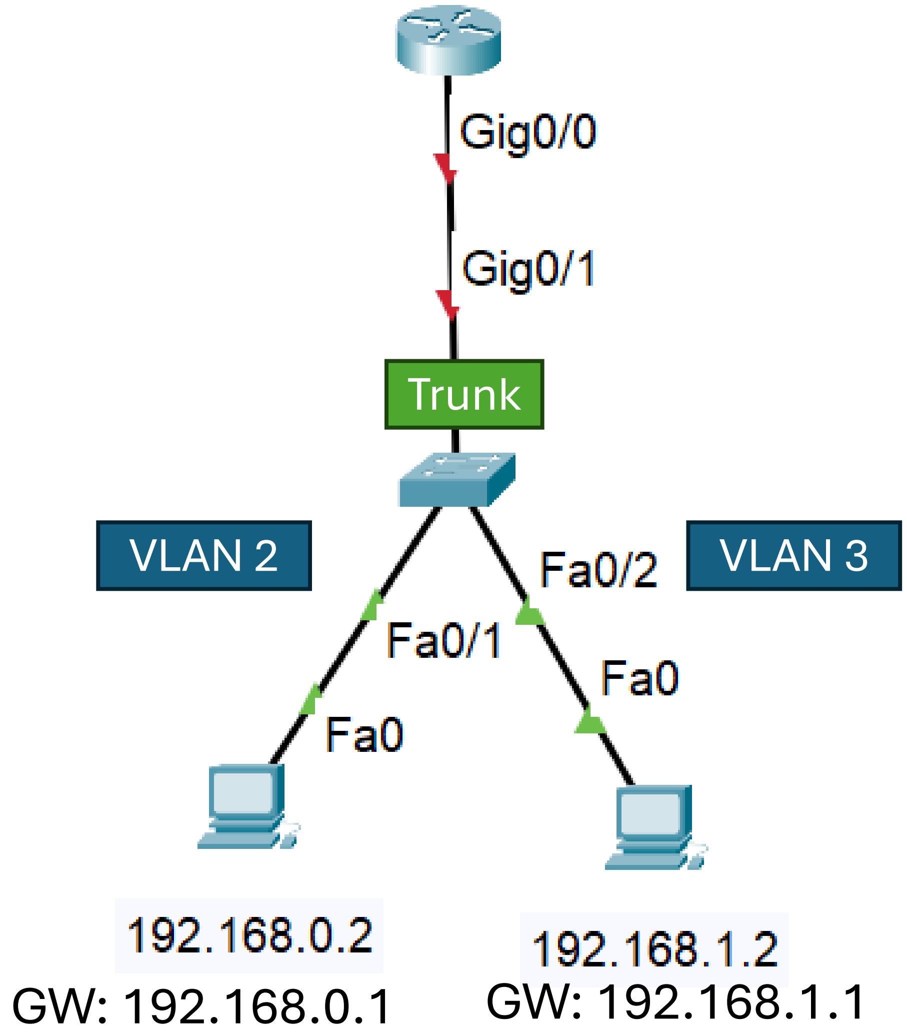
Uses

- Connecting two computers directly
- Connecting two switches directly
- Connecting two routers directly



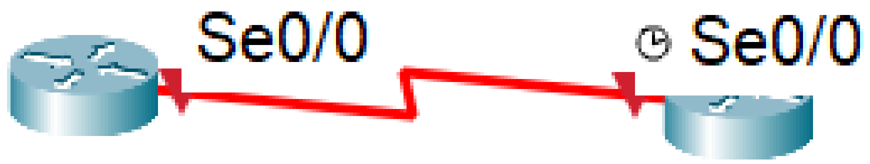
IntrVLAN Example

```
Switch(config)#int f0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
% Access VLAN does not exist. Creating vlan 2
Switch(config-if)#exit
Switch(config)#int f0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
% Access VLAN does not exist. Creating vlan 3
Switch(config-if)#exit
Switch(config)#int g0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#do write
Building configuration...
[OK]
```

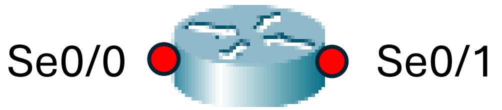


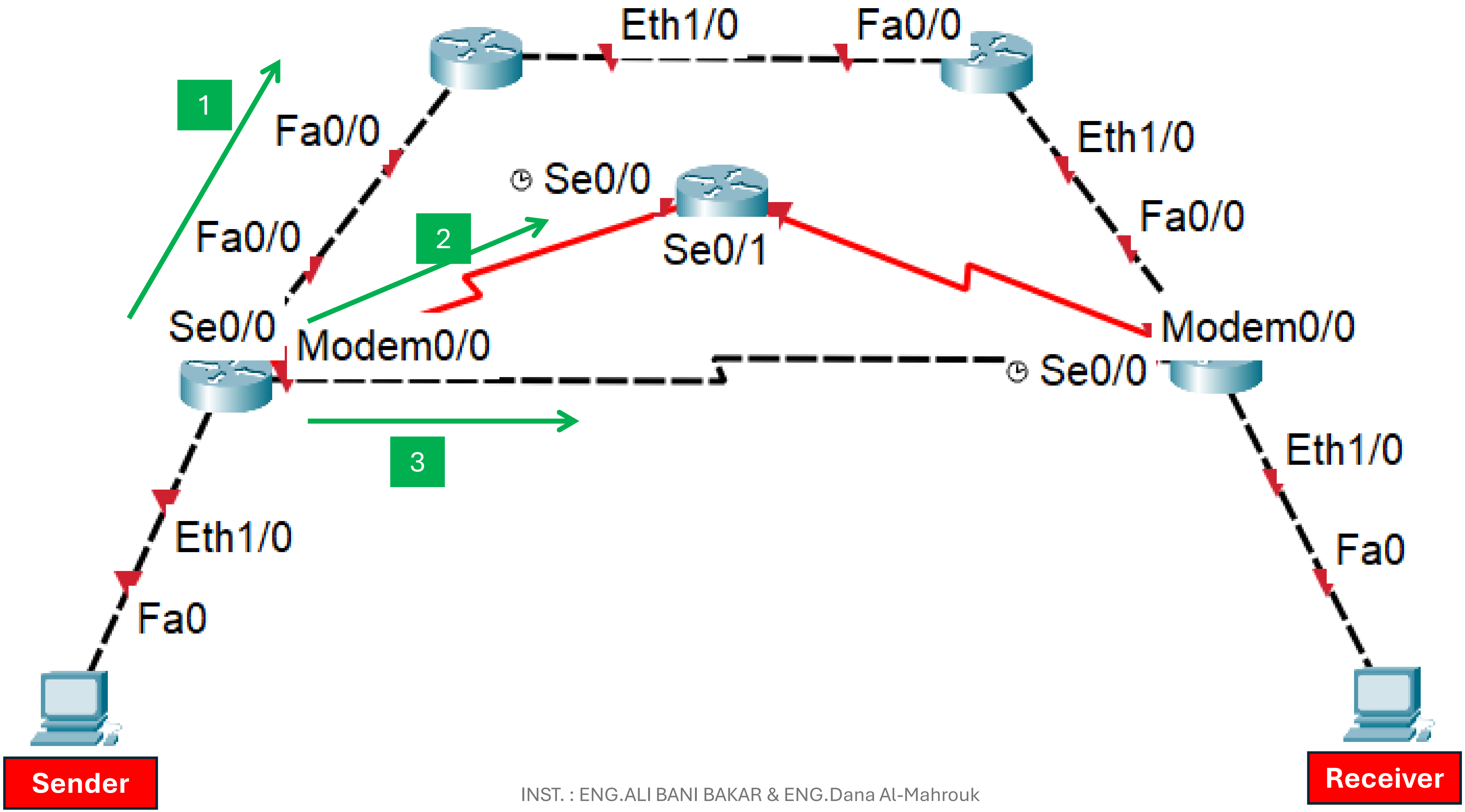
```
Router>en
Router#conf t
Router(config)#int g0/0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#int g0/0.1
Router(config-subif)#encapsulation dot1Q 2
Router(config-subif)#ip address 192.168.0.1 255.255.255.0
Router(config-subif)#exit
Router(config)#int g0/0.2
Router(config-subif)#encapsulation dot1Q 3
Router(config-subif)#ip address 192.168.1.1 255.255.255.0
Router(config-subif)#do wr
Building configuration...
[OK]
```

Routing

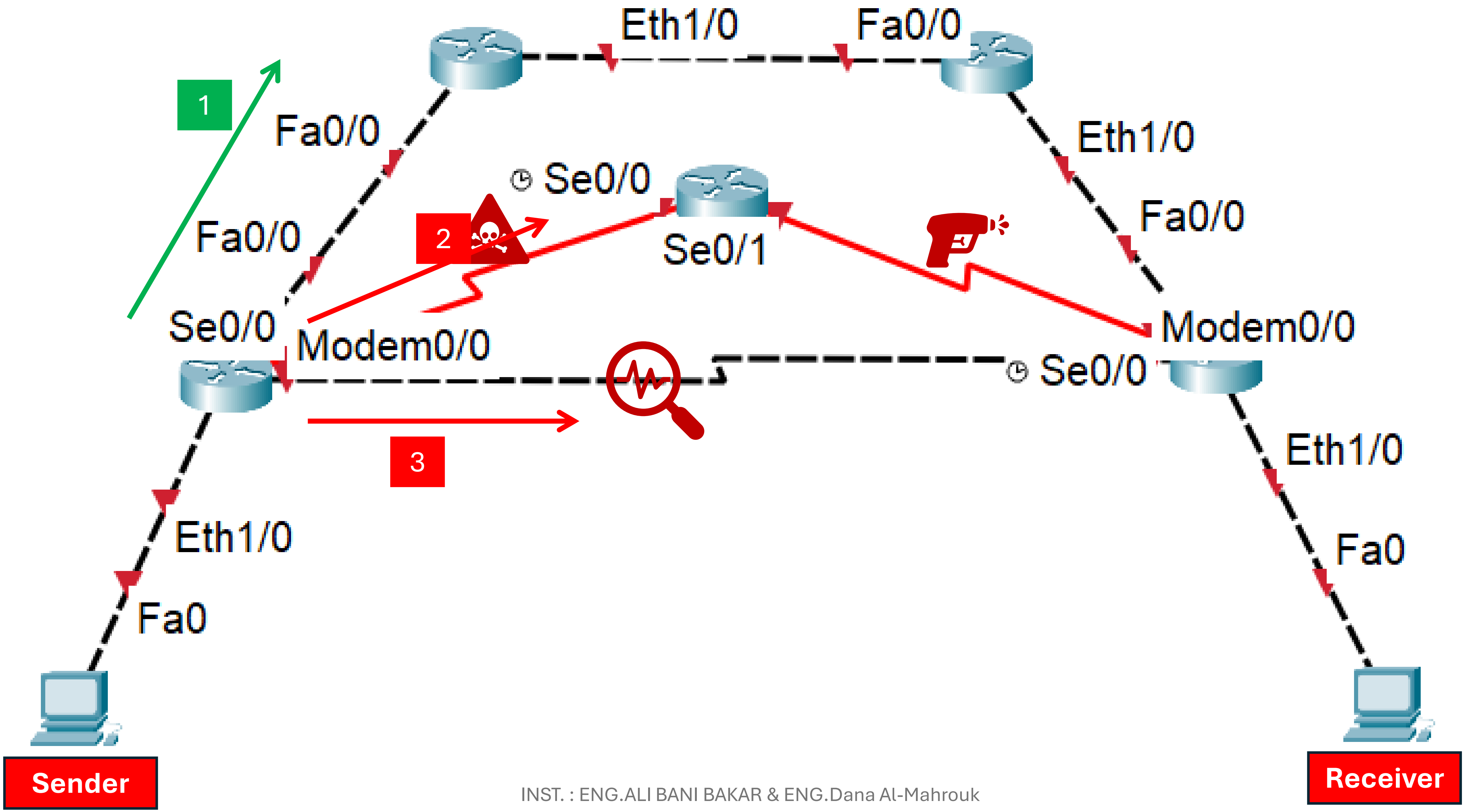


- From router to router, or from interface to interface
- Router Function:
  - Connection between different networks or VLANs.
  - Choose the best path (Manually or Dynamically)
  - DHCP, Telnet, and ping & so on





1
2
3
Fa0/0
Eth1/0
Fa0/0
Fa0/0
Se0/0
Se0/1
Eth1/0
Fa0/0
Se0/0
Modem0/0
Modem0/0
Se0/0
Eth1/0
Fa0
Eth1/0
Fa0
Sender
Receiver


Eth1/0
Fa0/0
1
Fa0/0
Se0/0
Fa0/0
2
Se0/1
Eth1/0
Fa0/0
Se0/0
Modem0/0
Modem0/0
Se0/0
3
Eth1/0
Fa0
Eth1/0
Fa0
Sender
Receiver

Static VS Dynamic Routing

	Static	Dynamic
Security	✓	✗
Performance (Speed)	✓	✗
Availability (Always On)	✗	✓
Ease of Configuration	✗	✓
Used	Small network & Secure	Large network

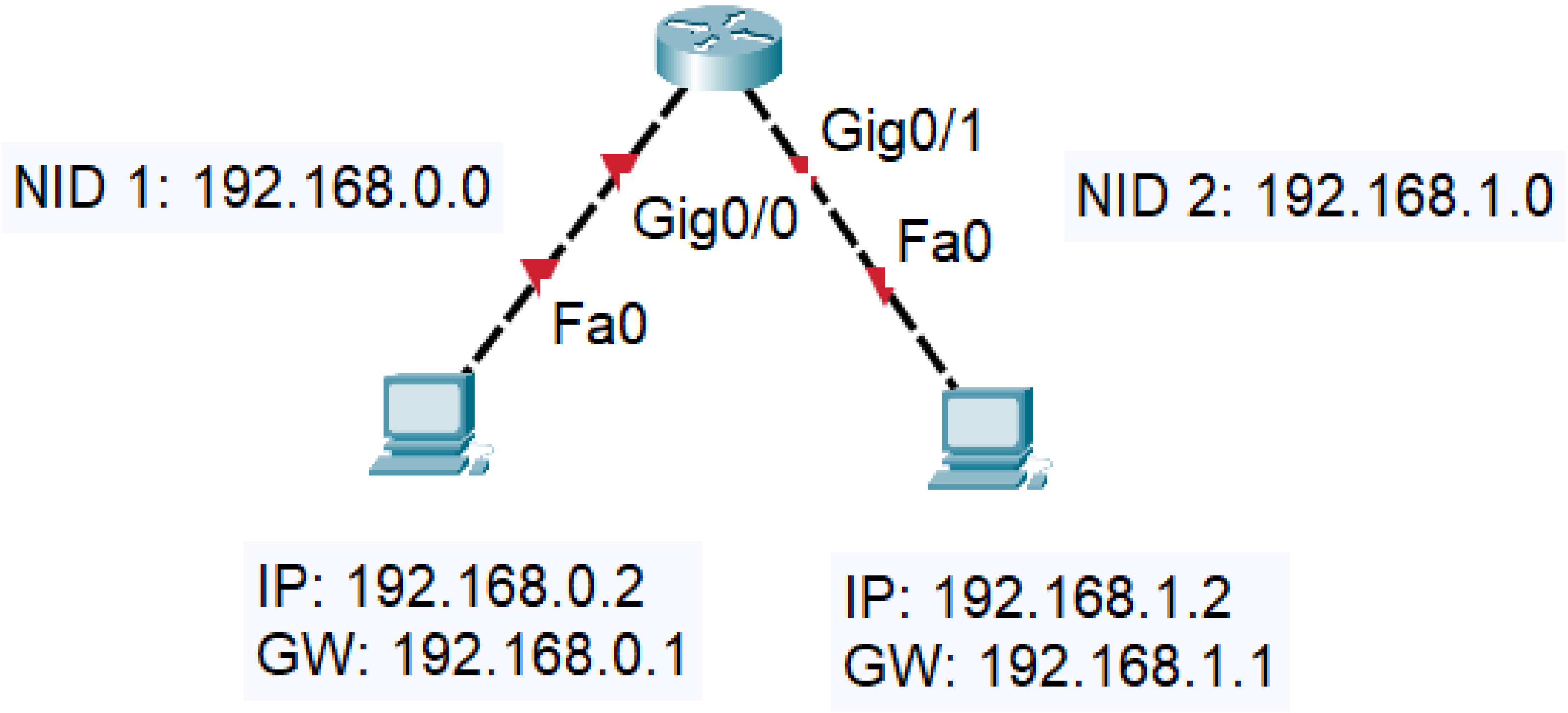
Static Routing Example

Router do not need a manual configuration if there are only one router.

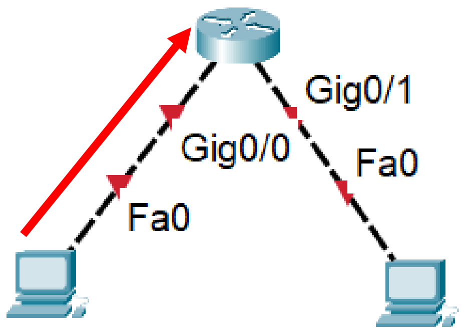
Because it have a Routing Table.

Router Table

Network ID	Interface
192.168.0.0	G0/0
192.168.1.0	G0/1

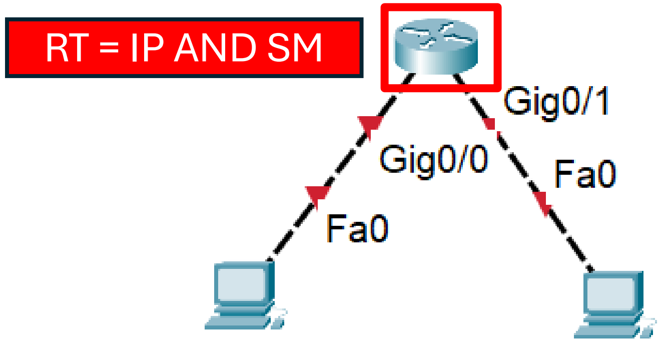


What Happens?



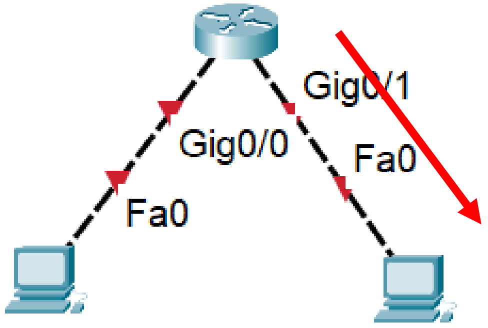
IP: 192.168.0.2
GW: 192.168.0.1

IP: 192.168.1.2
GW: 192.168.1.1



IP: 192.168.0.2
GW: 192.168.0.1

IP: 192.168.1.2
GW: 192.168.1.1



IP: 192.168.0.2
GW: 192.168.0.1

IP: 192.168.1.2
GW: 192.168.1.1

No configurations are placed on the router, it is **completely automatic**. When the message reaches the router, it sends it to the receiving network automatically, how does this happen?
The router contains a table called the **router table**, which contains the Network ID and the interface associated with it.

Network ID	Interface
192.168.0.0	G0/0
192.168.1.0	G0/1

When the message reaches the router, it does an ending between the dest IP address and the dest subnet mask.
Dest IP = 192.168.1.2
Dest SM = 255.255.255.0
Dest NID = 192.168.1.0
So… From Routing Table:
Int = G0/1

CLI

Router>en

Router#conf t

Router(config)#**int g0/0**

Router(config-if)#**no sh**utdown

Router(config-if)#**ip add**ress 192.168.**0.1** 255.255.255.0

Router(config-if)#exit

Router(config)#**int g0/1**

Router(config-if)#**no sh**utdown

Router(config-if)#**ip add**ress 192.168.**1.1** 255.255.255.0

Router(config-if)#do wr

Router(config-if)#exit

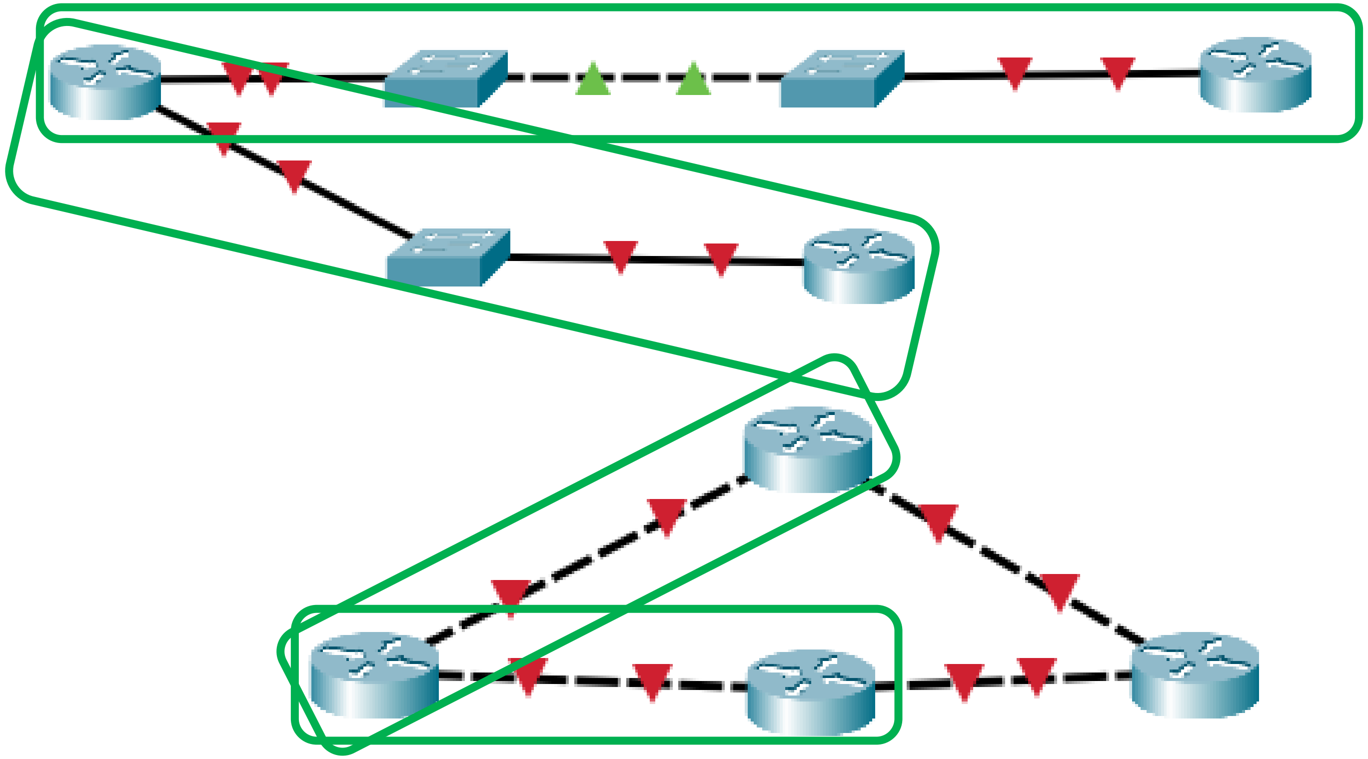
Routing Table

```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, GigabitEthernet0/0
L       192.168.0.1/32 is directly connected, GigabitEthernet0/0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/1
L       192.168.1.1/32 is directly connected, GigabitEthernet0/1
```

Direct Connect

- connecting two or more routers directly without the need for intermediary devices or networks.

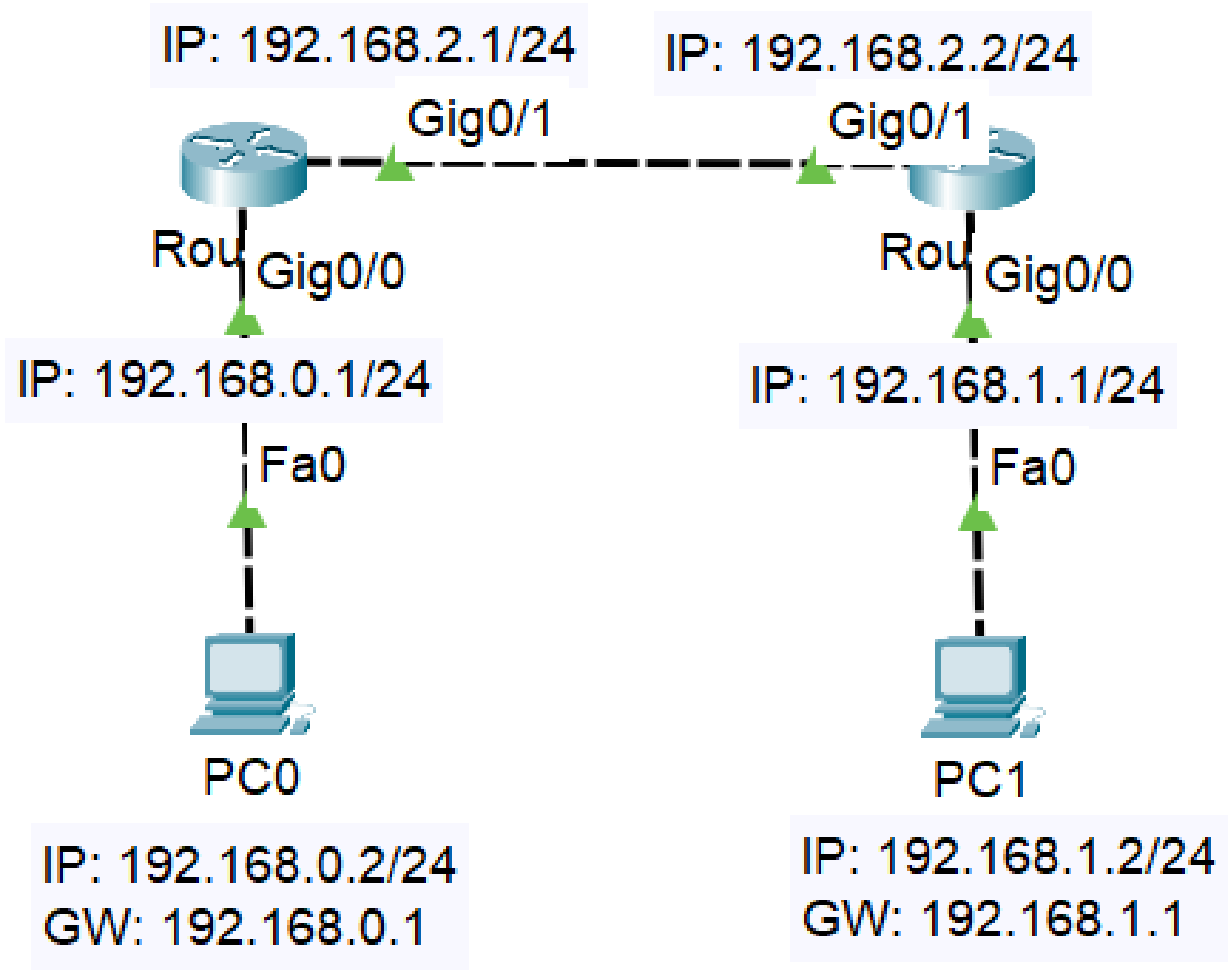
Direct connection is directly from the router to the next router, no matter how many switches are between them



Day 6

- Outline
  - Static Routing Example
  - Dynamic Routing (OSPF)
    - Area
    - Area Border Router
    - OSPF Projects
    - GNS3 Project
  - Router RAM, Flash, and NVRAM
  - Configuration Register (Rommon)

Static Routing Example



CLI

Router>en

Router#conf t

Router(config)#**int g0/0**

Router(config-if)#**no sh**utdown

Router(config-if)#**ip add**ress 192.168.**0.1** 255.255.255.0

Router(config-if)#exit

Router(config)#**int g0/1**

Router(config-if)#**no sh**utdown

Router(config-if)#**ip add**ress 192.168.**2.1** 255.255.255.0

Router(config-if)#do wr

Router(config-if)#exit

Router1>en

Router1#conf t

Router1(config)#**int g0/0**

Router1(config-if)#**no sh**utdown

Router1(config-if)#**ip add**ress 192.168.**1.1** 255.255.255.0

Router1(config-if)#exit

Router1(config)#**int g0/1**

Router1(config-if)#**no sh**utdown

Router1(config-if)#**ip add**ress 192.168.**2.2** 255.255.255.0

Router1(config-if)#do wr

Router1(config-if)#exit

Q: Is that enough to do ping?

No Connection...

Src IP	Dest IP	Data
192.168.0.2/24	192.168.1.2/24	Hi

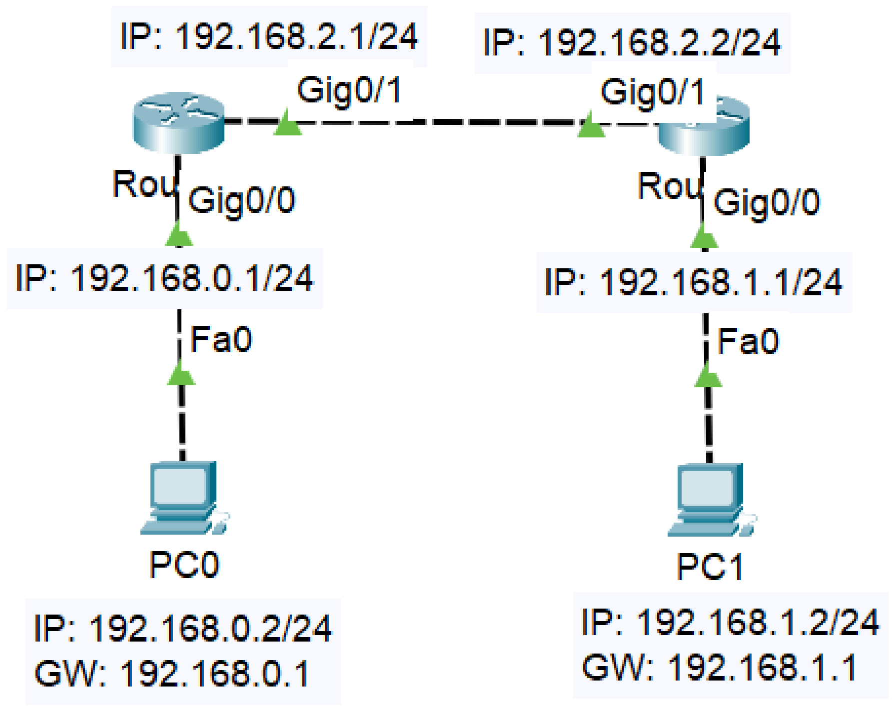
- When this message reaches the router, it finds that the receiving network is not in the table!!!

Dest IP = 192.168.1.2

Dest SM = 255.255.255.0

Dest NID = 192.168.1.0

➔ Dest Int = ?? Not found in R0 RT



R0 Router Table

Network ID	Interface
192.168.0.0	G0/0
192.168.2.0	G0/1

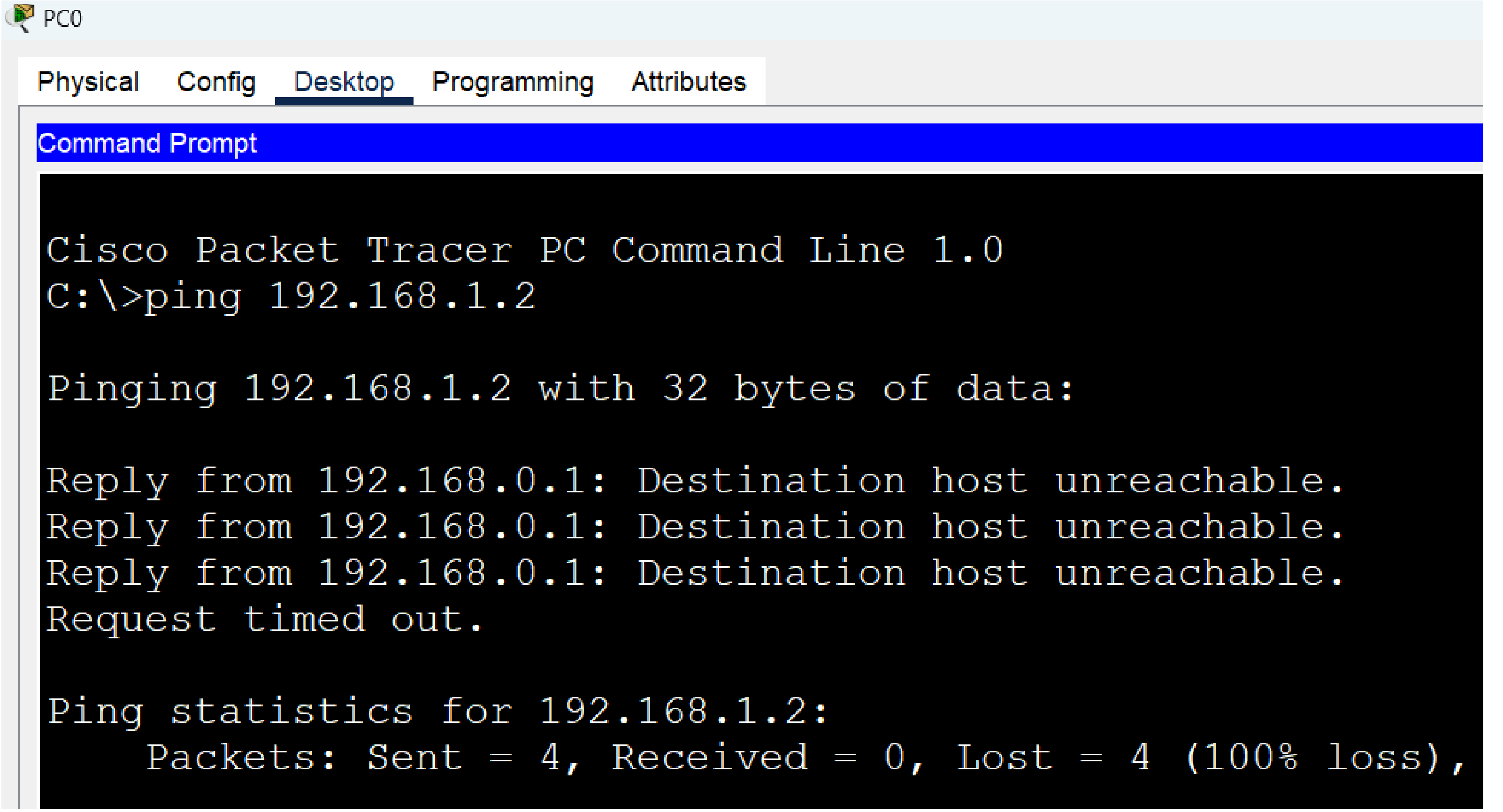
R1 Router Table

Network ID	Interface
192.168.1.0	G0/0
192.168.2.0	G0/1

Reply From the gateway…

- The interface sends a message to the sender that it did not find the requested network.

- Type of response:
  - Reply TTL (successfully connection)
  - Reply From Router Interface (dest NID is not found)
  - Request Time Out



How do we solve this problem?

- Now we just need to tell the first router (R0) that there is a network ID(192.168.1.0) and its interface is (G0/1), how do we do that?
- Static Way…

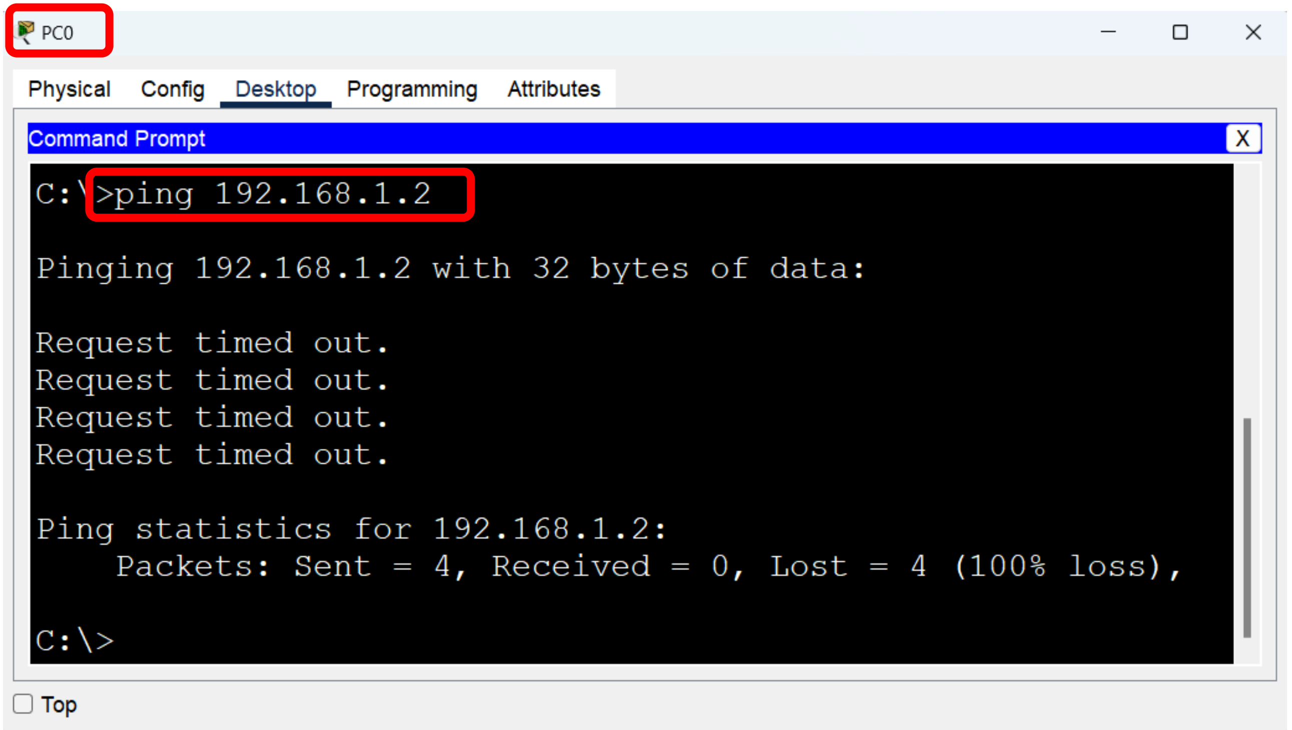
Router0

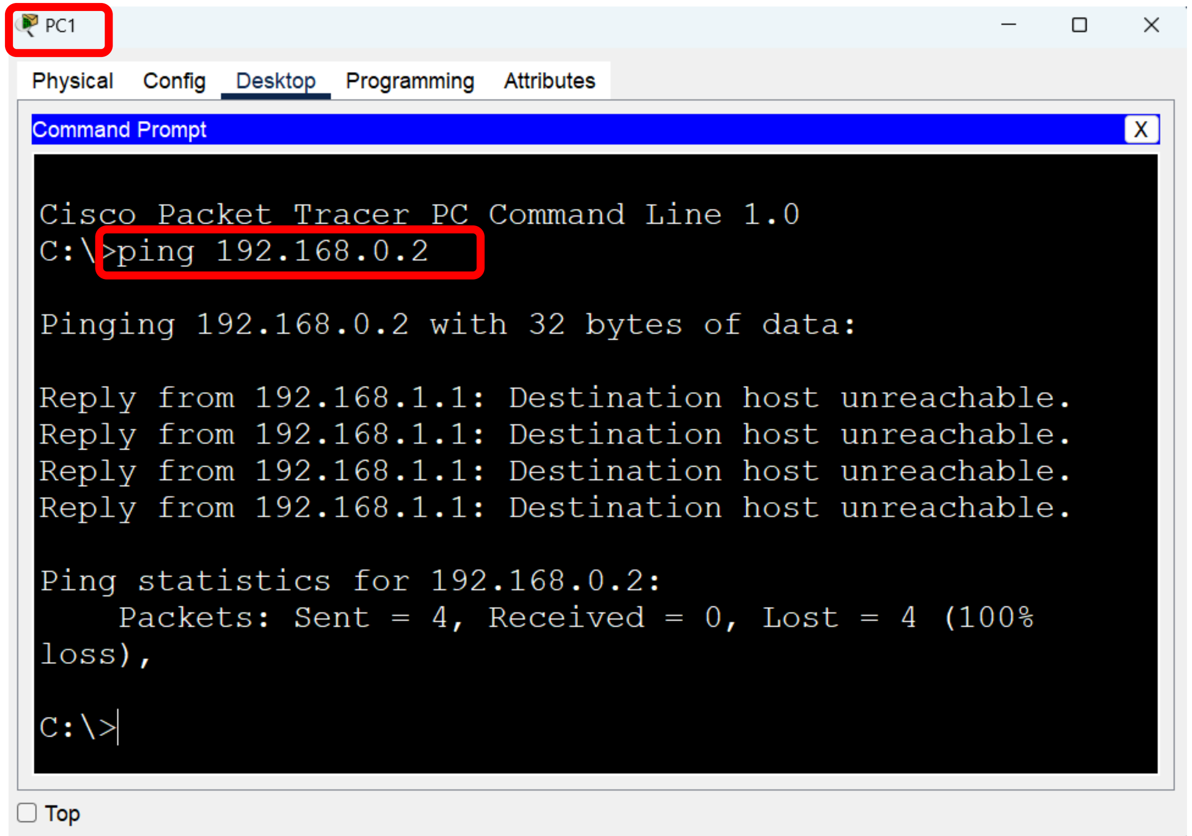
Physical Config CLI Attributes

IOS Command Line Interface

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 192.168.1.0 255.255.255.0 g0/1
%Default route without gateway, if not a point-to-point interface, may impact performance
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, GigabitEthernet0/0
L       192.168.0.1/32 is directly connected, GigabitEthernet0/0
S    192.168.1.0/24 is directly connected, GigabitEthernet0/1
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, GigabitEthernet0/1
L       192.168.2.1/32 is directly connected, GigabitEthernet0/1
```

Will the devices communicate now?





Why request time out?

- After the message was sent from the Sender PC, it arrived correctly to the Receiver PC, and when the receiver PC sent the reply message, the second router (R1) could not find the 192.168.0.0 network, so no reply reached the sender PC.

Router1

Physical Config CLI Attributes

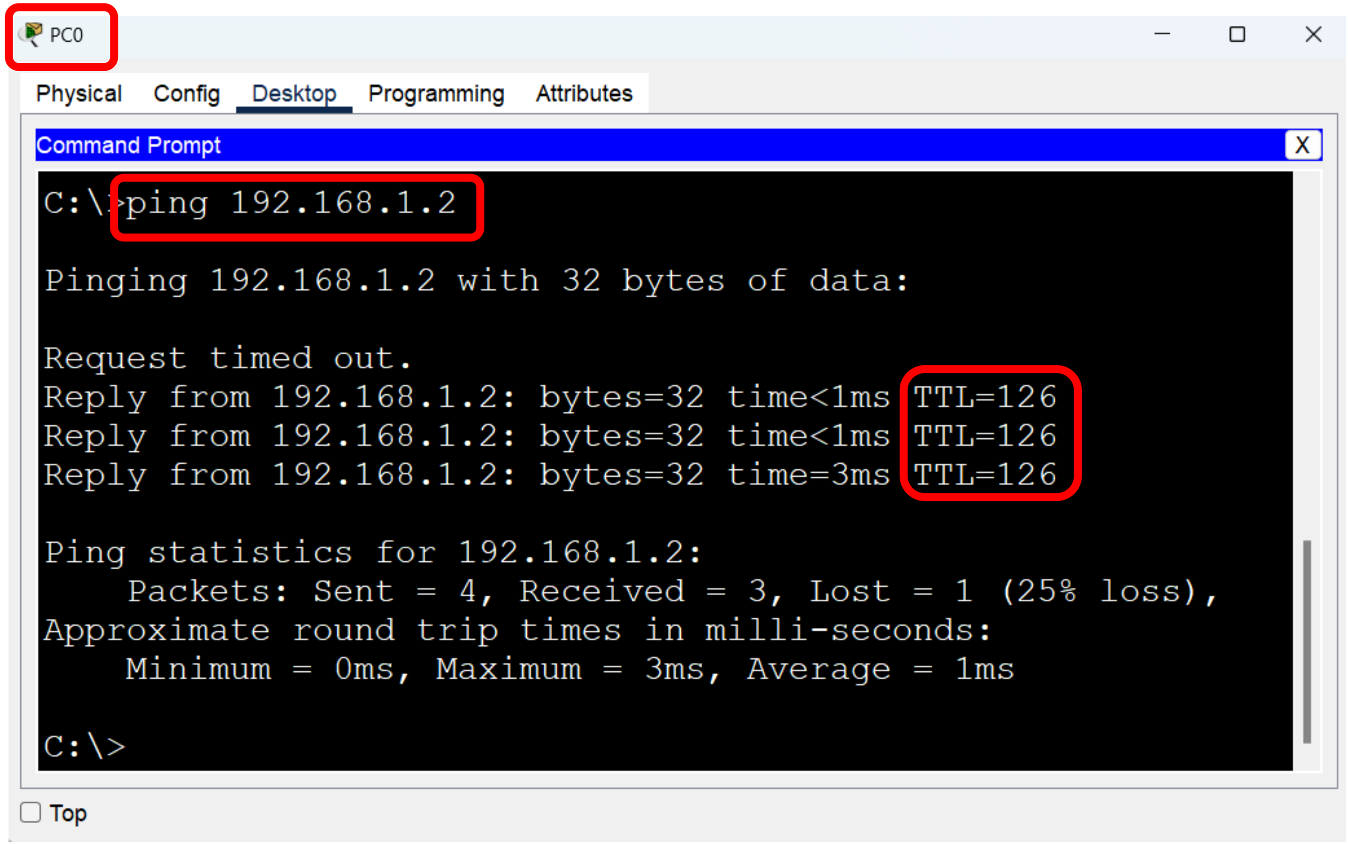
IOS Command Line Interface

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 192.168.0.0 255.255.255.0 g0/1
%Default route without gateway, if not a point-to-point interface, may impact performance
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

S    192.168.0.0/24 is directly connected, GigabitEthernet0/1
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, GigabitEthernet0/1
L       192.168.2.2/32 is directly connected, GigabitEthernet0/1
```

Ping Scanner

- Searching for TTL after pinging devices
- TTL is only present when the connection is successful

- If the connection is not successful or there is no response, this does not mean that the device is not present, this may be due to the work of the firewall, which in many cases prevents the response of messages to increase security



Routing

- Routing in a network involves determining the paths for data packets to travel from a source to a destination.
- Static Routing: involves **manually configuring** the routes in a network. This means that the **network administrator** specifies the paths that data packets should take to reach their destinations.

Advantages:

1. Simplicity: straightforward to implement in small networks.
2. Predictability: The path that packets will take is predictable and consistent.
3. Security: Less risk of routing loops or malicious route alterations.
4. Low Overhead: Reducing network overhead.

Disadvantages:

1. Lack of Scalability: Managing static routes becomes impractical in large, complex networks.
2. No Automatic Failover: If a route becomes unavailable, there is no automatic rerouting, which can lead to network downtime.

Dynamic Routing

- involves using routing protocols to **automatically discover and maintain the routes** within a network. Routers exchange routing information and update their routing tables dynamically as the network topology changes.

Advantages:

1. Scalability: suitable for large and complex networks with many routers and paths.

2. Automatic Failover: can detect changes in the network and reroute traffic automatically, enhancing network reliability and availability.

3. Ease of Maintenance: requires less manual intervention compared to static routing.

4. Adaptability: can adapt to changes in network topology, such as adding or removing routers and links.

Cont…

Disadvantages:

1. Complexity: more complex to configure and manage compared to static routing.

2. Resource Consumption: consume more CPU, memory, and bandwidth due to routing updates and calculations.

3. Potential for Routing Loops: can lead to routing loops and suboptimal routing.

Use Cases:

- Large Enterprise Networks: is ideal for large enterprises with multiple subnets and complex topologies.

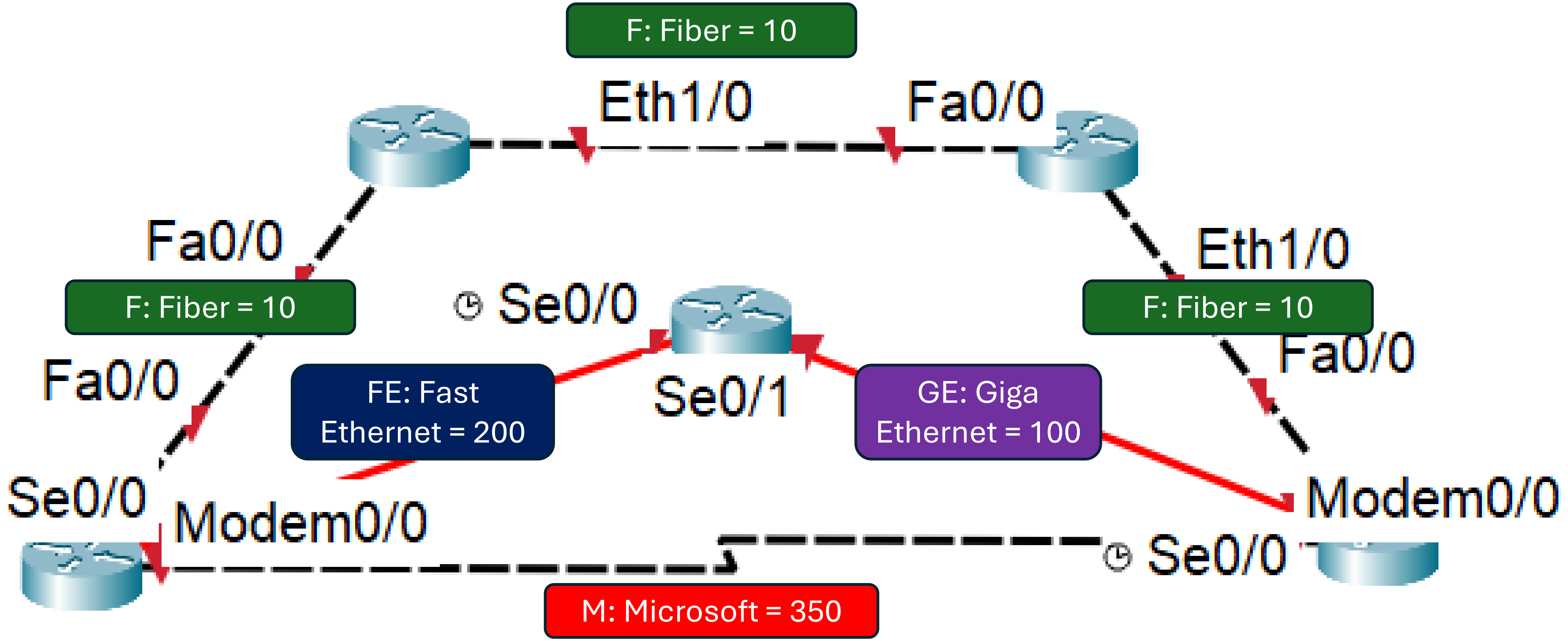
- Service Provider Networks: are commonly used by service providers to manage the routing of large-scale networks.

- Highly Available Networks: Networks that require high availability and automatic failover benefit from dynamic routing.

OSPF (Open Shortest Path First)

- is a dynamic routing protocol used in Internet Protocol (IP) networks.
- Shortest Path First Algorithm: OSPF uses **Dijkstra's algorithm** to calculate the shortest path tree for each route, ensuring efficient and optimal path selection.
- Neighbor Discovery: OSPF routers discover other OSPF routers on their directly connected networks using Hello packets. Routers become neighbors if they agree on certain parameters.

Cost



F: Fiber = 10
Eth1/0
Fa0/0
Fa0/0
F: Fiber = 10
Se0/0
Eth1/0
F: Fiber = 10
Fa0/0
Fa0/0
FE: Fast Ethernet = 200
Se0/1
GE: Giga Ethernet = 100
Se0/0
Modem0/0
Modem0/0
Se0/0
M: Microsoft = 350

OSPF Steps

1. Connection Router
2. Dijkstra's algorithm
3. Router Table
4. Startup

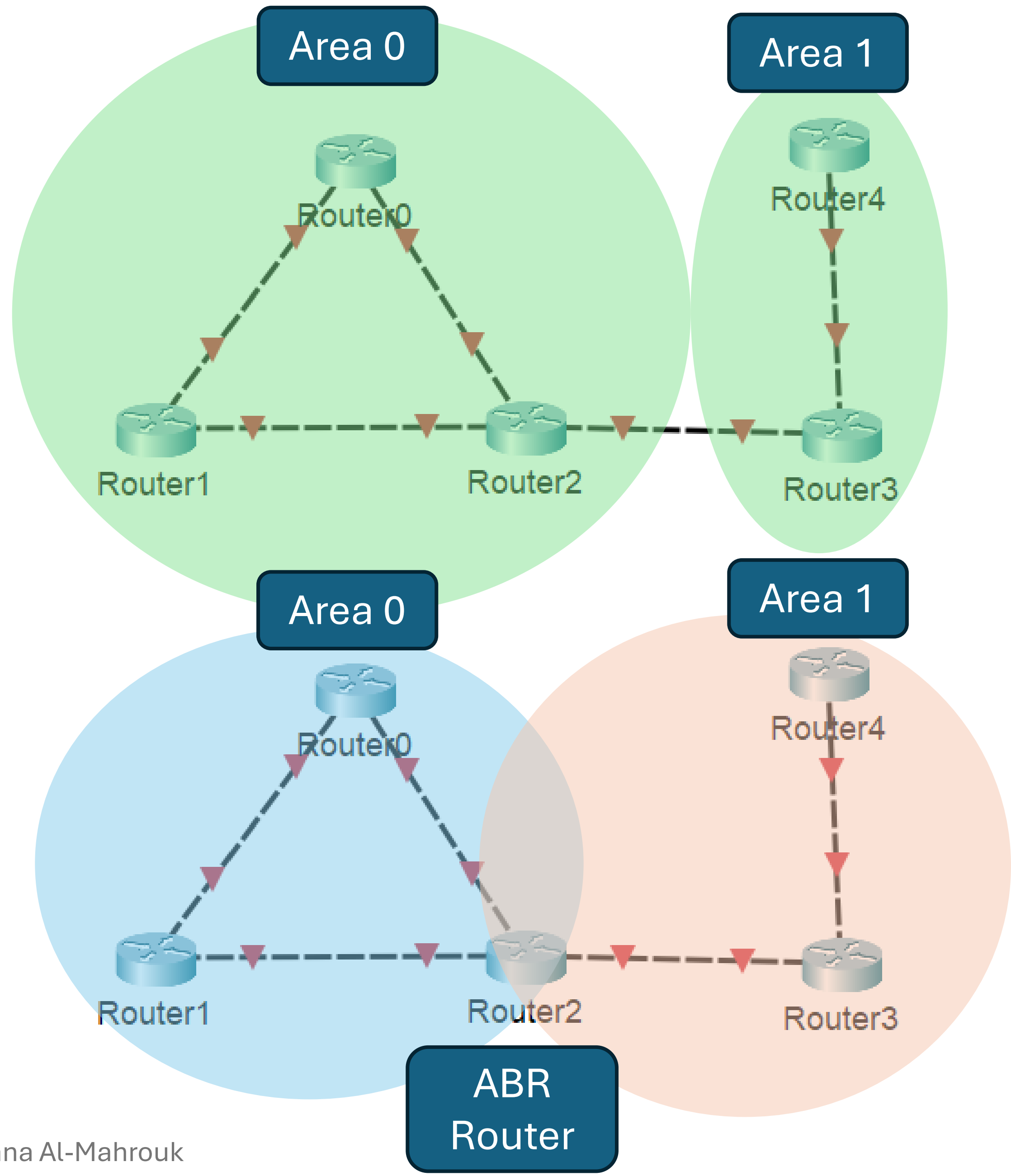
OSPF Area

- An OSPF area is a logical grouping of OSPF routers that helps manage routing by dividing a large network into smaller, more manageable segments. Each OSPF area maintains its own link-state database (LSDB) and limits the scope of LSAs (Link-State Advertisements) to reduce overhead and improve performance.

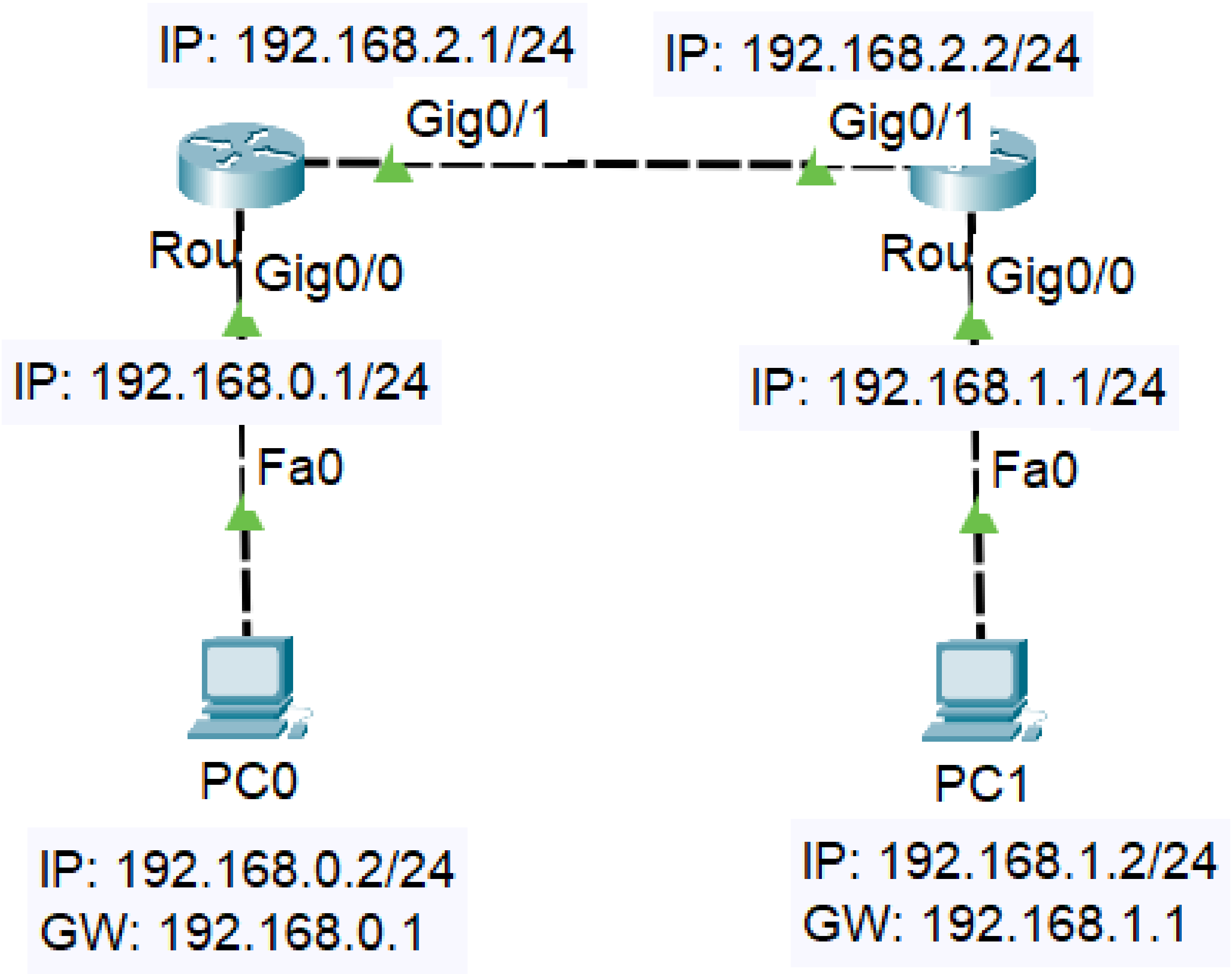
ABR Router

1. Internal Router: A router with all interfaces within the same OSPF area.

2. Area Border Router (ABR): Connects two or more OSPF areas, maintaining separate LSDBs for each area and propagating routing information between them.



Dynamic Routing Example 1



CLI

Router>en

Router#conf t

Router(config)#**int g0/0**

Router(config-if)#**no sh**utdown

Router(config-if)#**ip add**ress 192.168.**0.1** 255.255.255.0

Router(config-if)#exit

Router(config)#**int g0/1**

Router(config-if)#**no sh**utdown

Router(config-if)#**ip add**ress 192.168.**2.1** 255.255.255.0

Router(config-if)#do wr

Router(config-if)#exit

Router1>en

Router1#conf t

Router1(config)#**int g0/0**

Router1(config-if)#**no sh**utdown

Router1(config-if)#**ip add**ress 192.168.**1.1** 255.255.255.0

Router1(config-if)#exit

Router1(config)#**int g0/1**

Router1(config-if)#**no sh**utdown

Router1(config-if)#**ip add**ress 192.168.**2.2** 255.255.255.0

Router1(config-if)#do wr

Router1(config-if)#exit

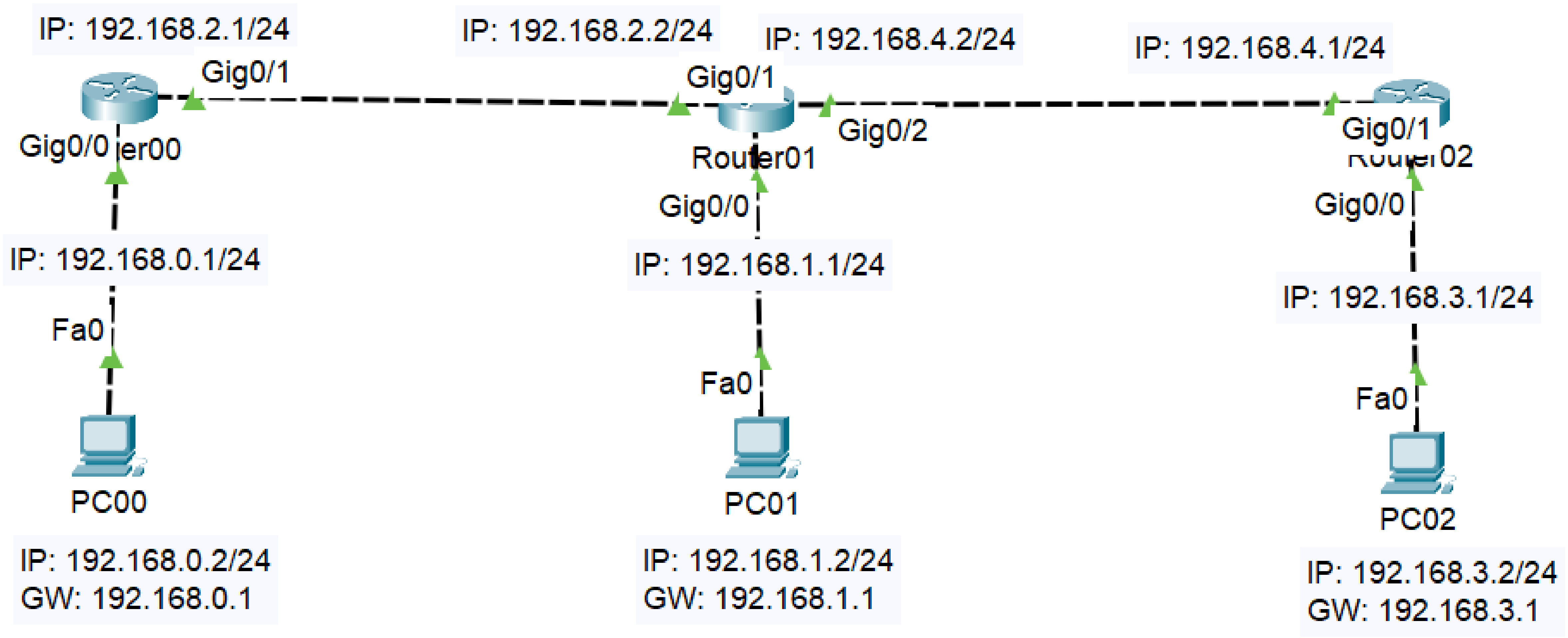
Q: Is that enough to do ping?

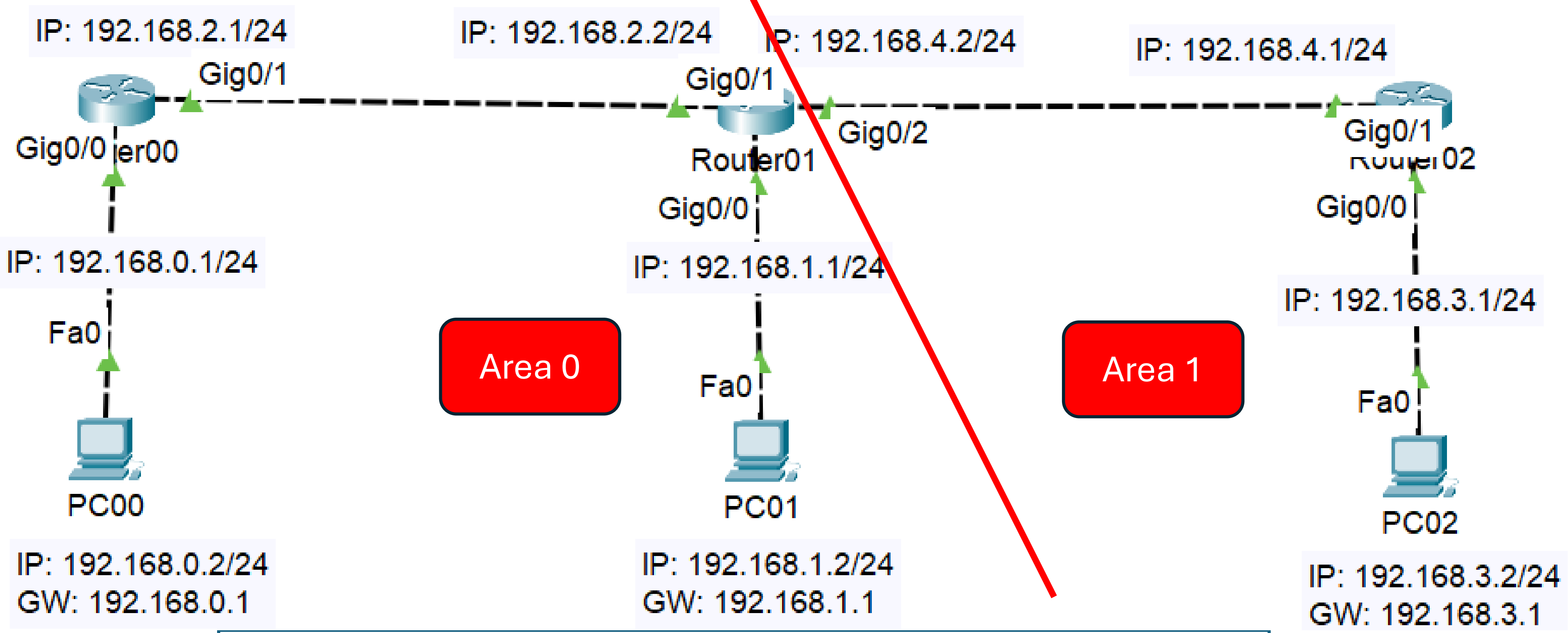
- This is the same example we used before, and we solve it using Static Routing.
- How to solve it now using Dynamic Routing ➔ OSPF Protocol??

```
Router(config)#route ospf 1
Router(config-route)# network  192.168.0.0  0.0.0.255  area 0
Router(config-route)# network  192.168.1.0  0.0.0.255  area 0
```

```
Router1(config)#route ospf 1
Router1(config-route)# network  192.168.2.0  0.0.0.255  area 0
Router1(config-route)# network  192.168.1.0  0.0.0.255  area 0
```

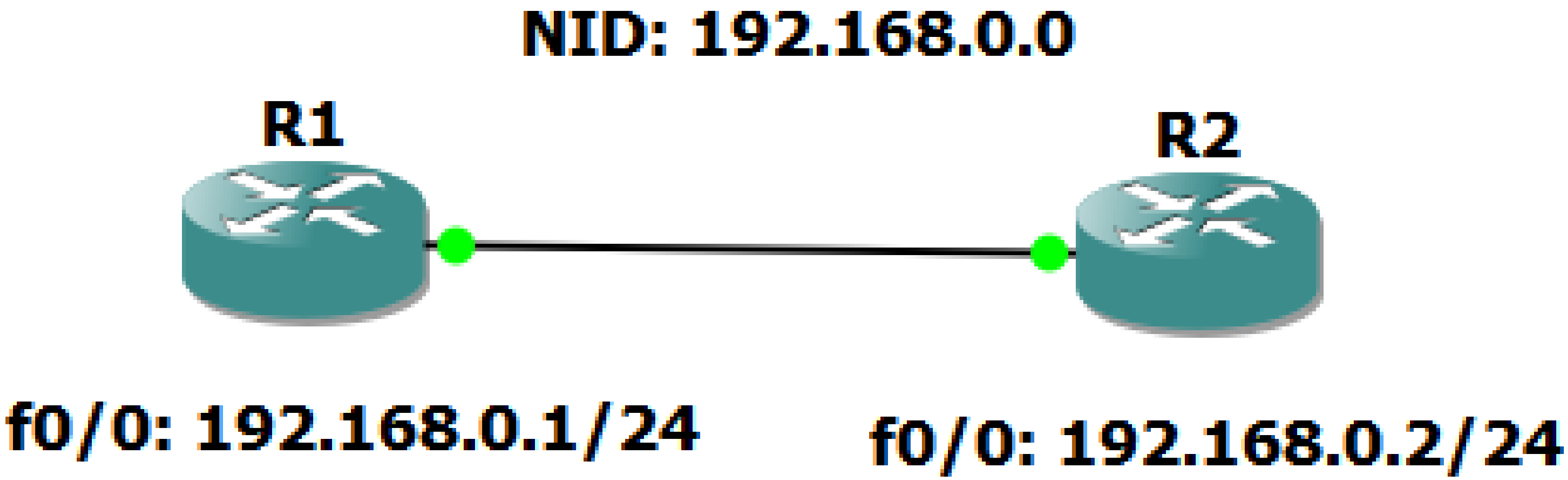
Dynamic Routing Example 2





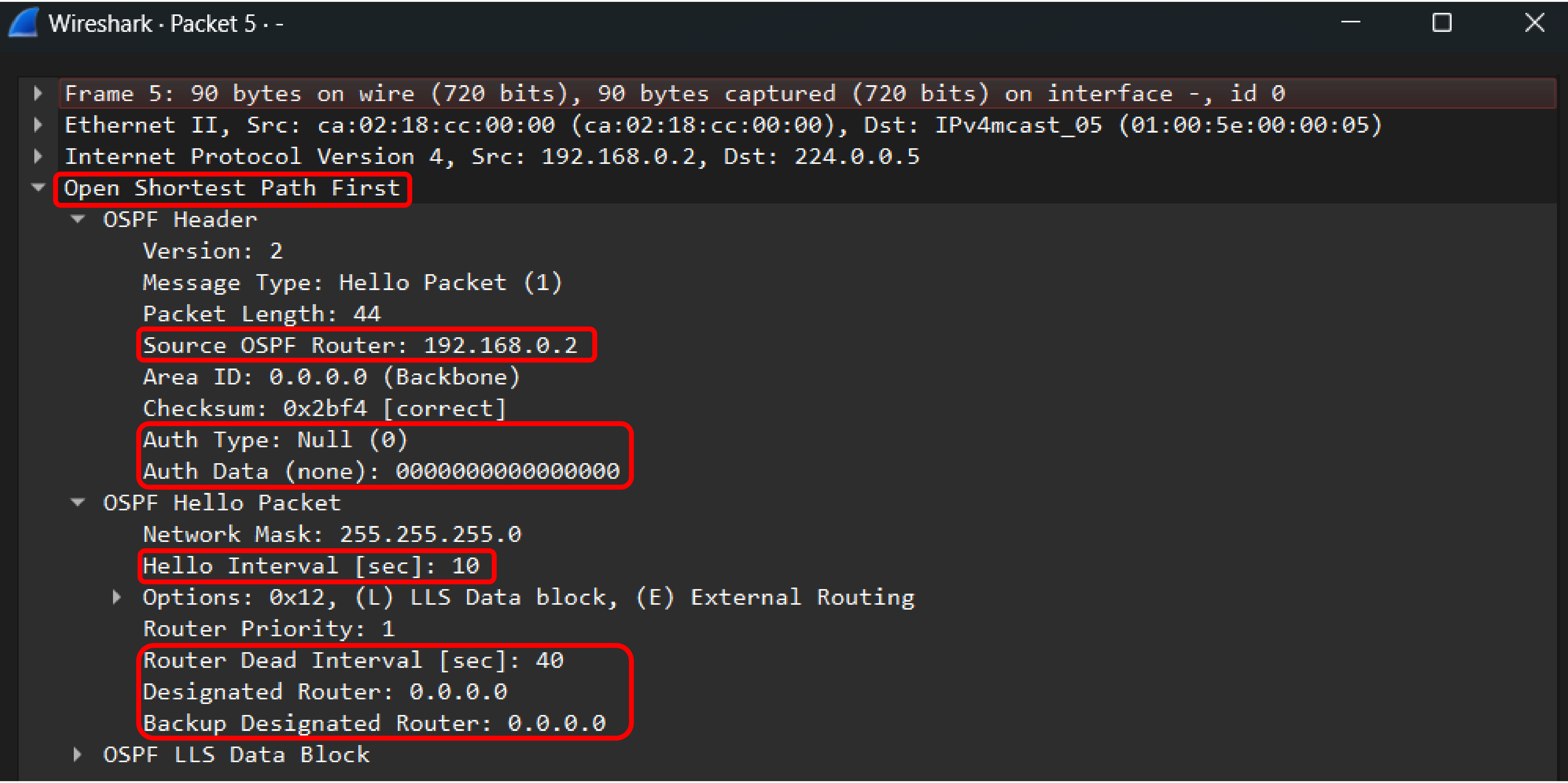
Router0(config)#**route ospf 1** Router0(config-route)# **network 192.168.0.0 0.0.0.255 area 0** Router0(config-route)# **network 192.168.2.0 0.0.0.255 area 0**
Router1(config)#**route ospf 1** Router1(config-route)# **network 192.168.1.0 0.0.0.255 area 0** Router1(config-route)# **network 192.168.2.0 0.0.0.255 area 0** Router1(config-route)# **network 192.168.4.0 0.0.0.255 area 1**
Router2(config)#**route ospf 1** Router2(config-route)# **network 192.168.4.0 0.0.0.255 area 1** Router2(config-route)# **network 192.168.3.0 0.0.0.255 area 1**

GNS3 Example

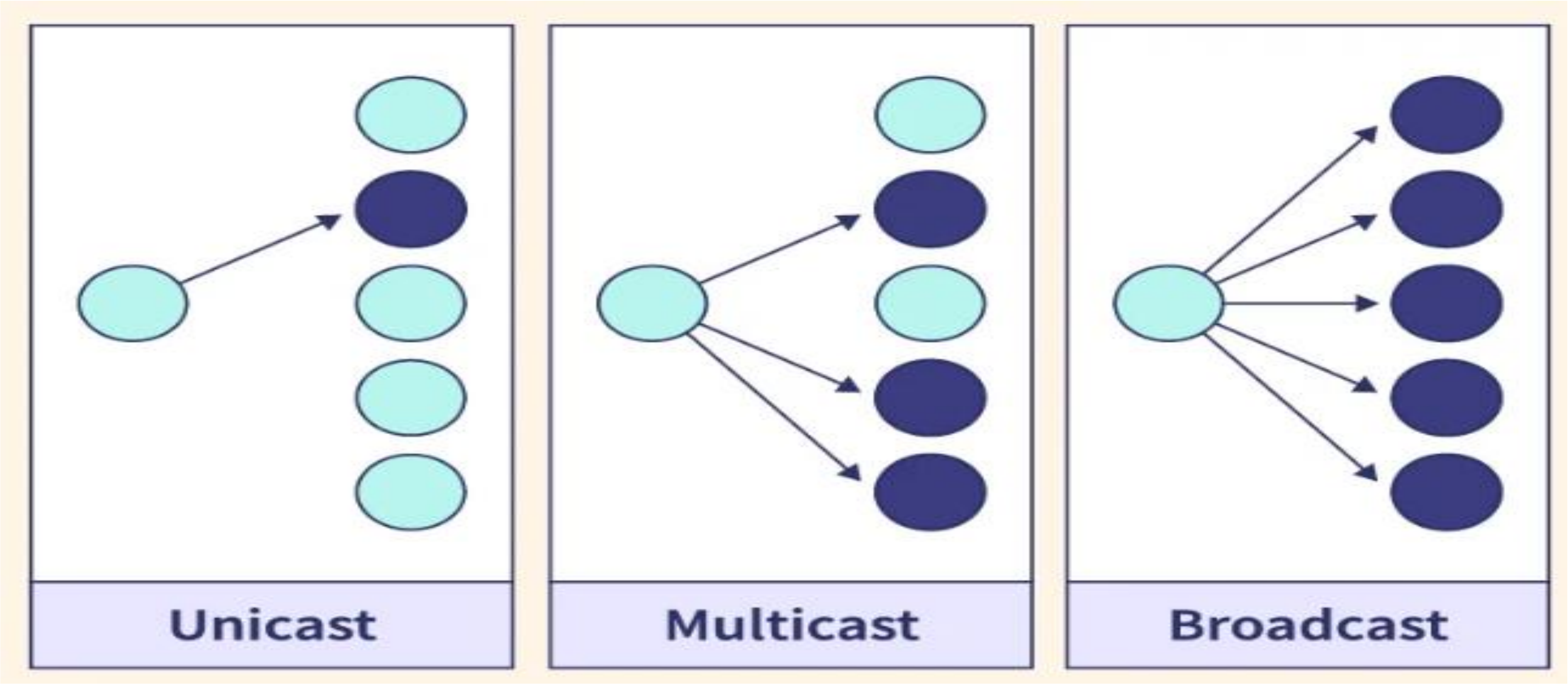


ospf

No.	Time	Source	Destination	Protocol	Length	Info
5	3.069785	192.168.0.2	224.0.0.5	OSPF	90	Hello Packet
11	12.505539	192.168.0.2	224.0.0.5	OSPF	90	Hello Packet
15	14.767526	192.168.0.1	224.0.0.5	OSPF	90	Hello Packet
19	21.684738	192.168.0.2	224.0.0.5	OSPF	94	Hello Packet
20	21.700359	192.168.0.1	192.168.0.2	OSPF	94	Hello Packet
22	24.746914	192.168.0.1	224.0.0.5	OSPF	94	Hello Packet
24	31.418094	192.168.0.2	224.0.0.5	OSPF	94	Hello Packet
26	34.464648	192.168.0.1	224.0.0.5	OSPF	94	Hello Packet
28	40.948355	192.168.0.2	224.0.0.5	OSPF	94	Hello Packet
29	43.025995	192.168.0.2	192.168.0.1	OSPF	78	DB Description
31	44.112619	192.168.0.1	224.0.0.5	OSPF	94	Hello Packet


Wireshark · Packet 5 · -
Frame 5: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface -, id 0
Ethernet II, Src: ca:02:18:cc:00:00 (ca:02:18:cc:00:00), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
Internet Protocol Version 4, Src: 192.168.0.2, Dst: 224.0.0.5
Open Shortest Path First
OSPF Header
Version: 2
Message Type: Hello Packet (1)
Packet Length: 44
Source OSPF Router: 192.168.0.2
Area ID: 0.0.0.0 (Backbone)
Checksum: 0x2bf4 [correct]
Auth Type: Null (0)
Auth Data (none): 0000000000000000
OSPF Hello Packet
Network Mask: 255.255.255.0
Hello Interval [sec]: 10
Options: 0x12, (L) LLS Data block, (E) External Routing
Router Priority: 1
Router Dead Interval [sec]: 40
Designated Router: 0.0.0.0
Backup Designated Router: 0.0.0.0
OSPF LLS Data Block

Message



Broadcast

Broadcasting involves sending a data packet from **one sender to all** possible recipients within a network segment.

Characteristics:

- **One-to-All**: The sender transmits the data to all devices on the network.
- **No Specific Target**: Every device on the network segment receives the broadcast packet.
- **Network Flooding**: Broadcast traffic can cause network congestion if used excessively.

Use Cases:

- **ARP Requests**: Used to discover the MAC address corresponding to an IP address.
- **DHCP Discovery**: Used by a client to find available DHCP servers on the network.
- **Routing Protocol Updates**: Some protocols use broadcasts to distribute routing information.

Uni-cast

Unicasting involves sending a data packet **from one sender to a specific recipient**.

Characteristics:

- **One-to-One**: The sender transmits the data to a single, specific recipient.

- **Direct Communication**: Only the intended recipient processes the unicast packet.

- **Efficient Use of Bandwidth**: No unnecessary data transmission to other devices.

Use Cases:

- **Standard Network Communication**: Most internet traffic (e.g., web browsing, file transfers) is unicast.

- **Email**: Sending an email from one user to another.

Multicast

Multicasting involves sending a data packet from **one sender to multiple specific recipients**, typically belonging to a multicast group.

Characteristics:

- One-to-Many: The sender transmits the data to multiple recipients that are part of a multicast group.

- Efficient Distribution: Data is only sent to devices that have joined the multicast group, reducing unnecessary traffic.

- Requires Support: Network infrastructure and devices must support multicast to handle it properly.

Running Config

RAM

OS Config

Startup Config

NVRAM

OS Configuration:

- password
- IP Address
- OSPF
- Hostname
- ~~No shutdown~~

Flash

~~OS~~

IOS (without configuration)

copy startup-config running-config

copy running-config startup-config → do write

Configuration Register

Configuration Register Bits Breakdown

- **Bits 0-3**: Boot Field
  - **0x0**: Manual boot (ROM Monitor mode)
  - **0x1**: Boot to the first image in the flash
  - **0x2-F**: Specifies a specific image to boot
- **Bits 6-7**: Console Speed
  - **00**: 9600 bps (default)
  - **01**: 4800 bps
  - **10**: 2400 bps
  - **11**: 1200 bps
- **Bit 8**: Break Enable/Disable
  - **0**: Break disabled (default)
  - **1**: Break enabled
- **Bit 10**: **Ignore NVRAM**
  - **0**: Use startup-config (default)
  - **1: Ignore startup-config (useful for password recovery)**

Configuration Register **0x2102**

0x2102 is the default configuration register setting for most Cisco routers.

Boot Normally: Load the IOS from flash memory.

Use NVRAM: Read the startup configuration file (usually located in NVRAM) to configure the router.

Baud Rate: Uses the default console baud rate of 9600 bps.

Configuration Register **0x2142**

- **Ignore NVRAM**: During boot, the router ignores the startup configuration file in NVRAM. This allows the router to boot without applying the saved configuration.
- This setting is primarily used for:

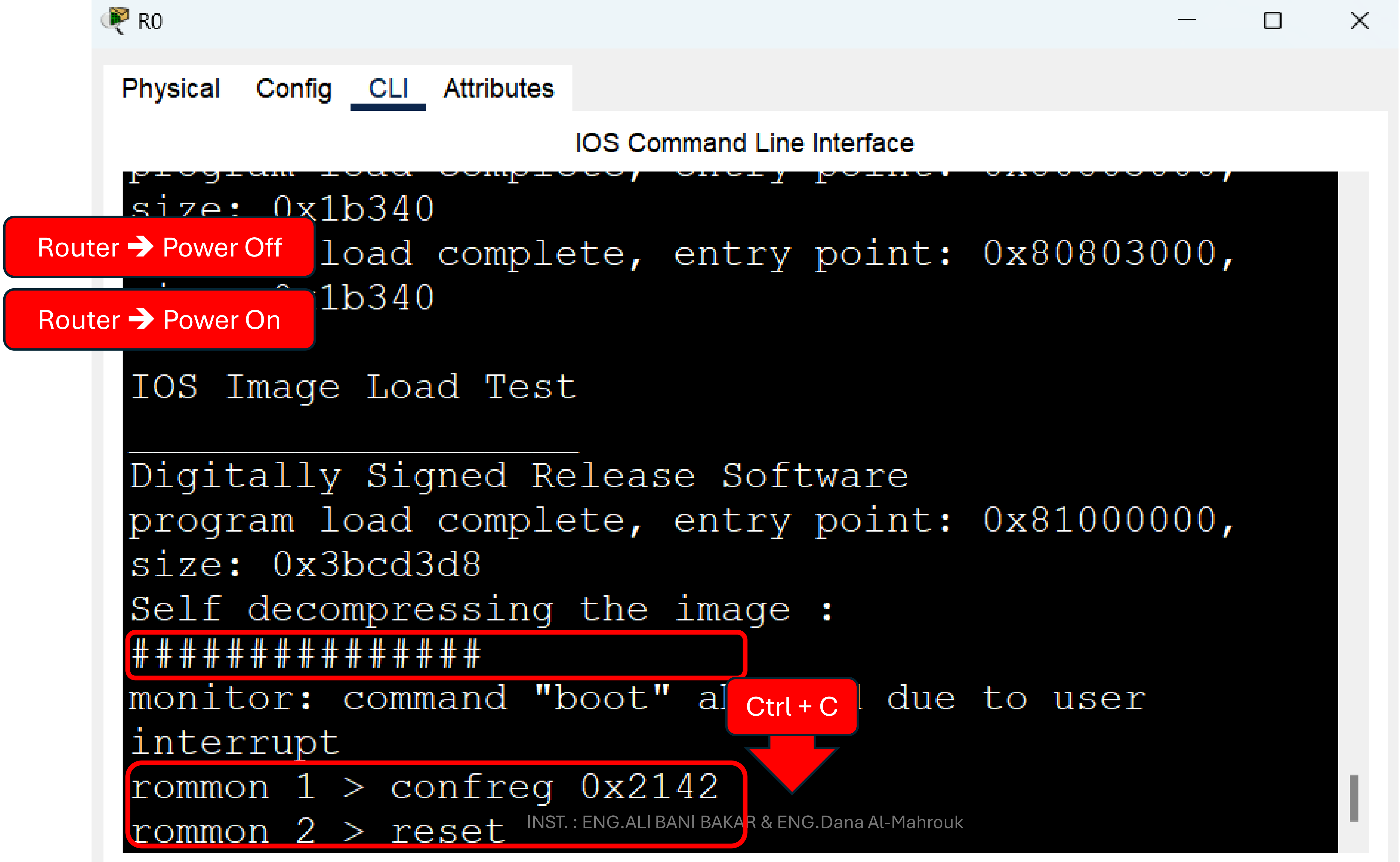
- **Password Recovery**: When an **administrator has forgotten the enable password or other critical access passwords**.
- **Troubleshooting**: If a faulty configuration is preventing the router from booting correctly, ignoring the startup configuration allows for troubleshooting and correcting the issue without applying the problematic configuration.

R0

Physical Config CLI Attributes

IOS Command Line Interface

```
Router#
Router#conf t
Enter configuration commands, one per line.  End
with CNTL/Z.
Router(config)#ho
Router(config)#hostname NAme_Royte_1
NAme_Royte_1(config)#do wr
Building configuration...
[OK]
NAme Royte 1(config)#enable pass
NAme_Royte_1(config)#enable password 122234
NAme_Royte_1(config)#do wr
Building configuration...
[OK]
NAme Royte 1(config)#
```


R0
Physical
Config
CLI
Attributes
IOS Command Line Interface
size: 0x1b340
load complete, entry point: 0x80803000,
1b340
IOS Image Load Test
Digitally Signed Release Software
program load complete, entry point: 0x81000000,
size: 0x3bcd3d8
Self decompressing the image :
###############
monitor: command "boot" a
due to user
interrupt
rommon 1 > confreg 0x2142
rommon 2 > reset
Router → Power Off
Router → Power On
Ctrl + C

```
Router#copy startup-config running-config
Destination filename [running-config]?
720 bytes copied in 0.416 secs (1730 bytes/sec)
NAme_Royte_1#
%SYS-5-CONFIG_I: Configured from console by console
NAme_Royte_1#conf t
NAme_Royte_1(config)#enable password 1234
NAme_Royte_1(config)#do wr
Building configuration...
[OK]
NAme_Royte_1(config)#config-register 0x2102
```

Thank You