Task 9.1p

Active class 9: Data-link Layer

Activity 1 involves understanding how devices in a Wi-Fi network communicate without interfering with each other. We simulate a Wi-Fi network where one person acts as the access point (AP) and others as devices. When someone wants to send data, they say "I'm sending a packet." If two people say this at the same time, there's a collision, and they need to try again. We then design a way to minimize these collisions and draw a timing diagram to illustrate our method. This helps us learn about the medium access control (MAC) protocols, like the one Wi-Fi uses, which helps devices take turns sending data to avoid collisions.

Activity 2 involves using Cisco Packet Tracer to simulate a network and understand how devices find each other. We assign static IP addresses to the devices and routers. We then discuss how a laptop (Laptop1) can find another computer (PC2) in the same network using the Address Resolution Protocol (ARP). ARP translates IP addresses into MAC addresses so devices can communicate. By pinging PC2 from Laptop1, we observe the ARP process, check the ARP tables, and compare them across different devices. This activity helps us understand how devices learn each other's addresses and communicate in a network.

Activity 3 focuses on how a switch manages and directs traffic within a network. We set up a network with a Cisco switch and configure devices with static IPs. We record the MAC addresses of all devices and the switch's Ethernet ports. By pinging between different devices, we observe how the switch updates its MAC address table to keep track of where each device is. We then clear the MAC table and switch to using DHCP for assigning IP addresses, comparing the new MAC and ARP tables with our initial observations. This activity teaches us how switches use MAC addresses to efficiently direct network traffic and how they adapt to changes in the network.

Notes

The Data Link Layer is like a bridge between the physical network and the higher layers of the network. It's responsible for making sure data gets from one device to another reliably across a physical connection, like Ethernet cables or Wi-Fi signals.

Addressing Devices: Every device connected to a network has a unique identifier called a MAC (Media Access Control) address. This layer uses MAC addresses to identify devices within the same network. It's like giving each device a name so they can talk to each other.

Spotting Errors: When data is sent, sometimes errors can happen along the way. The Data Link Layer helps detect these errors using methods like checksums. It's like adding a little code to the data to make sure it arrives intact.

Frame Structure: Before data is sent, it's divided into chunks called frames. Each frame has a header and a trailer added to it. These contain important information like where the data is going, error checking bits, and signals to help the devices sync up.

Sharing the Road: Imagine a bunch of cars trying to use the same road. The Data Link Layer decides who gets to go first and who has to wait, to avoid crashes (or in network terms, collisions). Different methods, like CSMA/CD for Ethernet, help devices take turns talking.

Controlling the Flow: It's like managing the speed of a river so it doesn't overflow. The Data Link Layer makes sure data is sent at a rate the receiving device can handle. This prevents one device from sending too much data too fast and overwhelming another device.

Logical Link Control: This part of the Data Link Layer focuses on establishing, maintaining, and terminating connections between devices. It ensures that data is delivered reliably and in the right order, even if there are multiple paths the data could take.

Common Protocols: Ethernet and Wi-Fi (802.11) are examples of protocols that operate at this layer. These protocols define how data is formatted, transmitted, and received over the network.

Hardware: The Data Link Layer operates at the hardware level, meaning it works directly with devices like switches and network interface cards (NICs). These devices handle the physical transmission of data within the network.

Examples: When you connect to the internet through a Wi-Fi router, the Data Link Layer helps ensure your data gets to and from your device and the router without getting mixed up with other devices' data. Similarly, in an Ethernet network, switches use the Data Link Layer to direct data only to the devices it's meant for, ensuring efficient communication within the network.

Error Detection and Avoidance Techniques:

Checksums: Adding a small piece of data to the message that allows the receiver to check for errors. If the checksum doesn't match what the receiver calculates, it knows there's an error.

Acknowledgment and Retransmission: After receiving data, the receiver sends back a message (acknowledgment) confirming it got the data correctly. If the sender doesn't receive this acknowledgment within a certain time, it assumes there was an error and sends the data again.

Frame Structure:

- **Header**: Contains information like source and destination MAC addresses, frame length, and control bits.
- **Data**: The actual data being transmitted.
- **Trailer**: Includes error-checking information like a checksum to ensure data integrity.

Flow Control:

Buffering: When a device receives data faster than it can process it, it stores the excess data in a buffer until it's ready to handle it. This prevents data loss or overflow.

Sliding Window Protocol: This technique allows the sender to keep sending data until it receives an acknowledgment from the receiver. It adjusts the window size dynamically based on network conditions to optimize data transmission.

Logical Link Control (LLC):

Purpose: LLC is responsible for establishing, maintaining, and terminating connections between devices. It ensures that data is delivered reliably and in the correct order, even if there are multiple paths the data could take.

Usage: LLC is part of the IEEE 802.2 standard and is often implemented in protocols like Ethernet and Token Ring.

Protocols Used and Their Purposes:

Ethernet (IEEE 802.3): Used for wired LANs, Ethernet defines how data is formatted and transmitted over the network. It's one of the most widely used protocols in the world.

Wi-Fi (IEEE 802.11): Used for wireless LANs, Wi-Fi allows devices to connect to a network without physical cables. It enables wireless communication between devices like laptops, smartphones, and routers.

Point-to-Point Protocol (PPP): Used for establishing a direct connection between two nodes over various physical mediums like serial cables, phone lines, or fiber optic cables. It's commonly used for internet connections over dial-up and DSL.

High-Level Data Link Control (HDLC): A bit-oriented protocol used for communication over point-to-point and multipoint links. It provides error detection and flow control.

Frame Relay: A packet-switching protocol used in Wide Area Networks (WANs) to transmit data between connected devices. It's known for its simplicity and efficiency in handling bursty traffic.

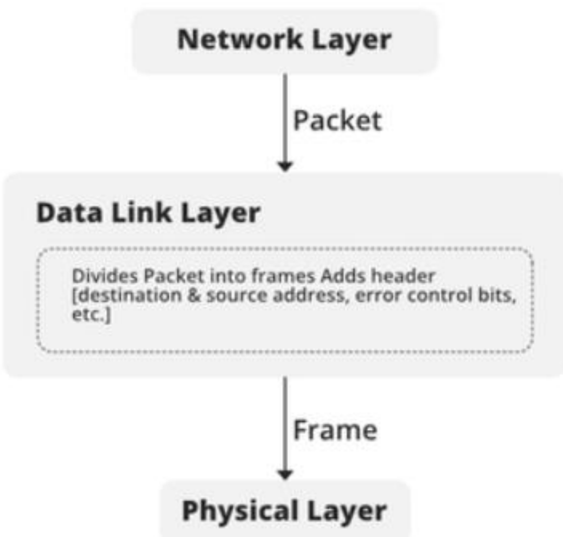
Asynchronous Transfer Mode (ATM): Used for transferring data, voice, and video over high-speed networks. It's based on transferring data in fixed-size cells, making it efficient for handling multimedia traffic.

These protocols serve different purposes but all operate within the Data Link Layer to ensure reliable and efficient communication within networks.

In summary, the Data Link Layer plays a crucial role in facilitating communication between devices within the same local network, ensuring data integrity, managing network resources, and establishing logical connections between network nodes. It serves as a bridge between the physical network medium and the higher layers of the OSI model, enabling seamless data transmission and network operation.

External resources I referred to

1. GeeksforGeeks. (2023, October 13). *Data link layer*. GeeksforGeeks. https://www.geeksforgeeks.org/data-link-layer/



2. Mitra, R., Brown, G., Huffman, M., & Zhu, H. (n.d.). *3. The data link layer*. Pressbooks. https://utsa.pressbooks.pub/networking/chapter/the-data-link-layer/

Quiz

Module 6: Data-Link Layer

Total points 100/100

Email *

rnirosh134@cicracampus.net

✓ Which of the following statement is true regarding reliable packet transfer? 10/10

- ○ If all the links in a network provide reliable frame delivery, then TCP's reliable delivery service can be replaced by the link layer's reliable service.
- ○ If all the links in a network provide reliable frame delivery, then TCP's reliable delivery service could be redundant.
- ◉ Even if all the links in a network provide reliable frame delivery, we still need to use TCP's reliable delivery service to guarantee the reliable delivery and also the in order delivery of the packets. ✓
- ○ Even if all the links in a network provide reliable frame delivery, we still need TCP's reliable delivery service to detect errors in the packets.

✓ Which of the following services are common in all link-layer protocols, TCP and IP? 10/10

- ◉ Framing and error detection ✓
- ○ framing and multiple access control
- ○ Error detection and multiple access control
- ○ Framing and reliable delivery

What is the most popular multiple access protocol used in WiFi? 10/10

- [x] CSMA/CA ✓
- [ ] ALOHA
- [ ] Pooling-based protocol
- [ ] Slotted-based protocol
- [ ] CSMA/CD

MAC address is 6 bytes long. How big is the MAC address space? 10/10

- [ ] 1099511627776
- [x] 281474976710656 ✓
- [ ] 64
- [ ] 4294967296
- [ ] 1024

Assume that we have three nodes connected to the same broadcast LAN. These nodes are Node X, Node Y, and Node Z. If Node Y sends a few hundreds of IP datagrams to Node X with each encapsulating frame with the destination address of Node X' MAC address

10/10

1) Will Node Z's adapter process these frames?

2) Will Node Z's adapter pass the IP datagrams encapsulated in these frames to Node Z's network layer?

- (selected, correct) Yes, No
- No, No
- Yes, Yes
- None of the answers are correct

Assume that we have three nodes connected to the same broadcast LAN. These nodes are Node X, Node Y, and Node Z. If Node Y sends a few hundreds of IP datagrams to Node Z with each encapsulating frame addressed to the **MAC broadcast address**,

10/10

1) Will Node X's adapter process these frames?

2) Will Node X's adapter pass the IP datagrams encapsulated in these frames to Node X's network layer?

- No, No
- Yes, No
- (selected, correct) Yes, Yes
- Can't determine with the information provided

Consider the following statements. 10/10

(a) ARP query sent in a broadcast frame

(b) ARP response sent within a frame with a particular destination MAC address.
Can you determine whether these two statements are true or false?

- (a) False (b) True
- (a) False (b) False
- (a) True (b) True ✓
- (a) True (b) False

The MAC address table is used by the Ethernet switches to store information about the MAC addresses associated with each physical port. How do Switches learn MAC addresses of its connected devices? 10/10

- Using the source MAC address. ✓
- Using the destination MAC address.
- Using DHCP
- Using ARP

We have a packet with a bit pattern 1010 0100 1101 1111 and suppose we use an even two-dimensional parity scheme. What would the value of the field containing the parity bits? 10/10

10100
01001
11011
11110
11000

Option 1 ✓

10100
01001
11011
11110
11100

Option 2

10100
01001
11011
11110
11110

Option 3

10100
01001
11011
11111
11000

Option 4

Assume that the router has three ARP modules, each with its own ARP table. Is it possible that the same MAC address appears in both tables? 10/10

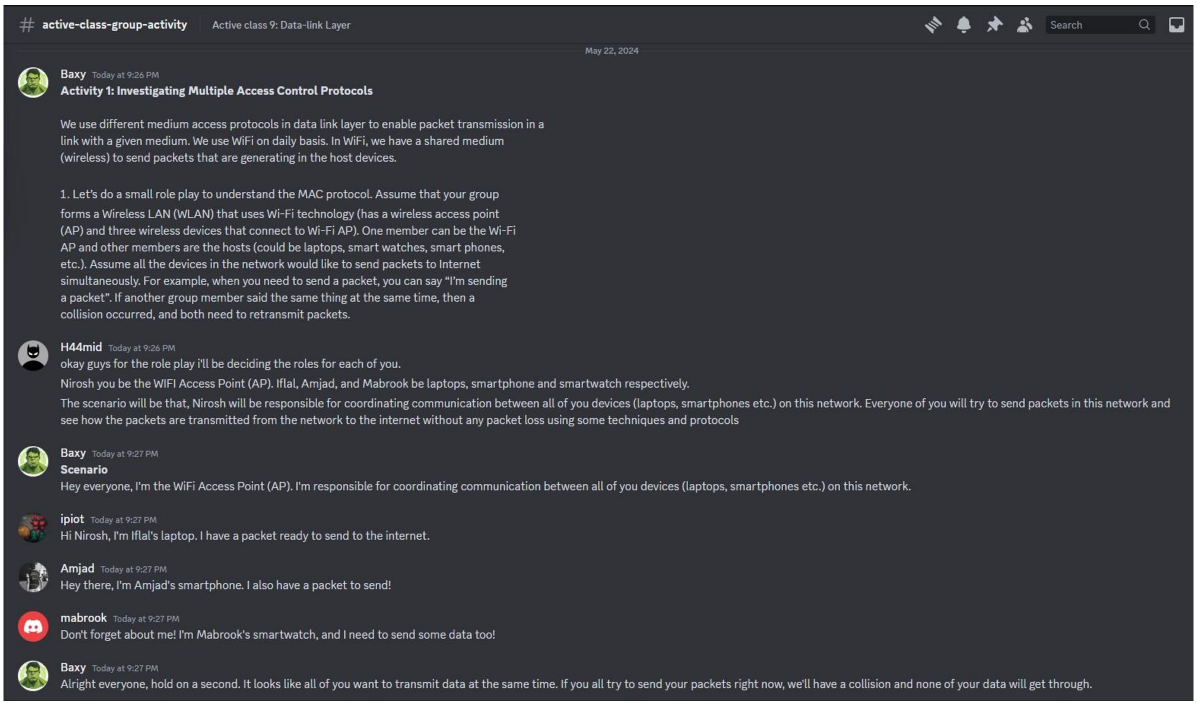
- No ✓
- Cannot determine
- Yes

Activity 1: Investigating Multiple Access Control Protocols

In this activity, we'll be acting out a WiFi network! I'll be the access point (AP), and others will be devices like laptops or phones.

Imagine we all want to send data at the same time. Since WiFi is like a shared conversation, we need a plan to avoid garbled messages. We'll design a system where we check if the "airwaves" are free before talking. If two devices try to talk at once, we'll use a random waiting time to avoid interrupting each other again. We'll also draw a timeline to show how this waiting time works. Finally, we'll learn about the real system WiFi uses to manage communication.

Our Discussion



active-class-group-activity | Active class 9: Data-link Layer

May 22, 2024

Baxy Today at 9:26 PM
Activity 1: Investigating Multiple Access Control Protocols

We use different medium access protocols in data link layer to enable packet transmission in a link with a given medium. We use WiFi on daily basis. In WiFi, we have a shared medium (wireless) to send packets that are generating in the host devices.

1. Let's do a small role play to understand the MAC protocol. Assume that your group forms a Wireless LAN (WLAN) that uses Wi-Fi technology (has a wireless access point (AP) and three wireless devices that connect to Wi-Fi AP). One member can be the Wi-Fi AP and other members are the hosts (could be laptops, smart watches, smart phones, etc.). Assume all the devices in the network would like to send packets to Internet simultaneously. For example, when you need to send a packet, you can say "I'm sending a packet". If another group member said the same thing at the same time, then a collision occurred, and both need to retransmit packets.

H44mid Today at 9:26 PM
okay guys for the role play i'll be deciding the roles for each of you.
Nirosh you be the WIFI Access Point (AP). Iflal, Amjad, and Mabrook be laptops, smartphone and smartwatch respectively.
The scenario will be that, Nirosh will be responsible for coordinating communication between all of you devices (laptops, smartphones etc.) on this network. Everyone of you will try to send packets in this network and see how the packets are transmitted from the network to the internet without any packet loss using some techniques and protocols

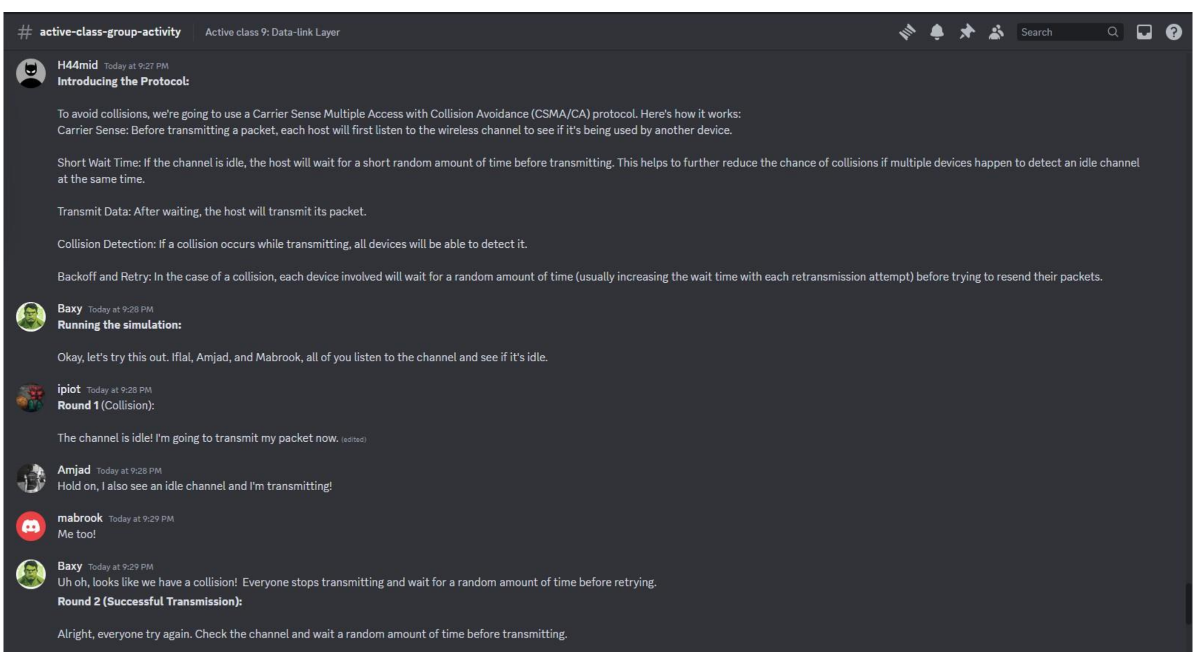
Baxy Today at 9:27 PM
Scenario
Hey everyone, I'm the WiFi Access Point (AP). I'm responsible for coordinating communication between all of you devices (laptops, smartphones etc.) on this network.

ipiot Today at 9:27 PM
Hi Nirosh, I'm Iflal's laptop. I have a packet ready to send to the internet.

Amjad Today at 9:27 PM
Hey there, I'm Amjad's smartphone. I also have a packet to send!

mabrook Today at 9:27 PM
Don't forget about me! I'm Mabrook's smartwatch, and I need to send some data too!

Baxy Today at 9:27 PM
Alright everyone, hold on a second. It looks like all of you want to transmit data at the same time. If you all try to send your packets right now, we'll have a collision and none of your data will get through.



active-class-group-activity | Active class 9: Data-link Layer

H44mid Today at 9:27 PM
Introducing the Protocol:

To avoid collisions, we're going to use a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. Here's how it works:
Carrier Sense: Before transmitting a packet, each host will first listen to the wireless channel to see if it's being used by another device.

Short Wait Time: If the channel is idle, the host will wait for a short random amount of time before transmitting. This helps to further reduce the chance of collisions if multiple devices happen to detect an idle channel at the same time.

Transmit Data: After waiting, the host will transmit its packet.

Collision Detection: If a collision occurs while transmitting, all devices will be able to detect it.

Backoff and Retry: In the case of a collision, each device involved will wait for a random amount of time (usually increasing the wait time with each retransmission attempt) before trying to resend their packets.

Baxy Today at 9:28 PM
Running the simulation:

Okay, let's try this out. Iflal, Amjad, and Mabrook, all of you listen to the channel and see if it's idle.

ipiot Today at 9:28 PM
Round 1 (Collision):

The channel is idle! I'm going to transmit my packet now. (edited)

Amjad Today at 9:28 PM
Hold on, I also see an idle channel and I'm transmitting!

mabrook Today at 9:29 PM
Me too!

Baxy Today at 9:29 PM
Uh oh, looks like we have a collision! Everyone stops transmitting and wait for a random amount of time before retrying.
Round 2 (Successful Transmission):

Alright, everyone try again. Check the channel and wait a random amount of time before transmitting.

active-class-group-activity Active class 9: Data-link Layer

ipiot Today at 9:29 PM
The channel is clear! I'm transmitting my packet.
(Mabrook and Amjad wait silently) (edited)

Baxy Today at 9:30 PM
Great job Iflal, your packet looks good! I'm sending it to the internet now.

Amjad Today at 9:30 PM
Round 3 (Another Transmission):

The channel is idle now. I'm going to transmit my packet.

Baxy Today at 9:30 PM
Perfect Amjad, your packet came through clearly. I'm sending it on its way.

mabrook Today at 9:31 PM
(Mabrook transmits)
Okay, my turn. The channel is clear and I've waited a bit. Here comes my packet!

Baxy Today at 9:31 PM
Excellent Mabrook, your packet looks good and I'm sending it off now.

There you go! By using CSMA/CA, we were able to avoid collisions and transmit everyone's data packets successfully.

2. What is the medium access control (MAC) protocol that can be used in WiFi?

This is a simplified example of CSMA/CA, and there are more complex variations used in Wi-Fi like CSMA/CA with Collision Detection (CSMA/CA with CD) that can further improve efficiency.

ipiot Today at 9:32 PM
That makes sense! Is CSMA/CA the only protocol used in Wi-Fi?

Baxy Today at 9:32 PM
That's a great question! There are actually other protocols used in Wi-Fi, but CSMA/CA is a common foundation for many of them. Think of them as teammates that help us manage the network traffic smoothly.

Amjad Today at 9:32 PM
Cool! Can you give us some examples?

active-class-group-activity Active class 9: Data-link Layer

Baxy Today at 9:32 PM
Sure! One is a set of standards called IEEE 802.11. Imagine these as different versions of Wi-Fi, like having different playbooks for different games. Each version, like 802.11ac or 802.11ax, specifies how we handle things like data speed, how many devices can communicate at once, and which frequency bands we use (think of them as different courts on which we play). All these variations work within the framework of CSMA/CA, but they add extra plays to our game.

mabrook Today at 9:32 PM
Interesting! Are there any other teammates on the field?

Baxy Today at 9:32 PM
Absolutely! Sometimes, even with CSMA/CA, collisions can happen. To minimize that risk, we can use a handshake called Request to Send/Clear to Send (RTS/CTS). It's like raising your hand before speaking in class. The device wanting to transmit asks the access point for permission (sends an RTS message). If the channel is clear, the access point gives the green light (sends a CTS message). This way, everyone knows who's talking and avoids interrupting each other.

ipiot Today at 9:32 PM
That makes sense! Are there any other tricks up our sleeves?

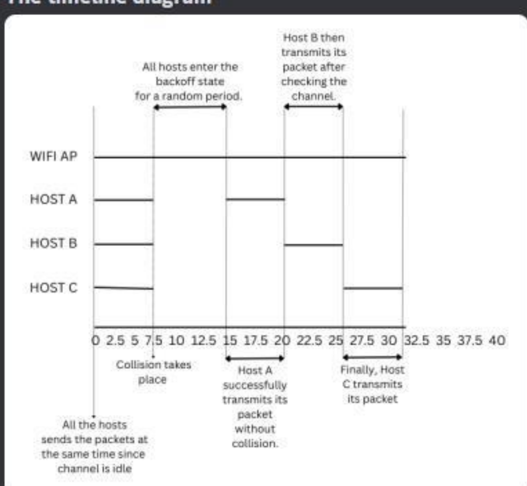
Baxy Today at 9:32 PM
You bet! Large data packets can be a bit cumbersome to transmit, so we can break them down into smaller pieces (fragmentation) before sending them. This reduces the risk of collisions and keeps things moving smoothly. On the other hand, if we have many small packets, we can combine them (aggregation) for better efficiency. It's like grouping smaller tasks together to save time.

Amjad Today at 9:33 PM
Whoa, that's some complex teamwork!

Baxy Today at 9:33 PM
It can be, but it all works together to make Wi-Fi a reliable and efficient way for us to connect and share data. There are even more protocols used depending on the situation, but these are some of the key players.

mabrook Today at 9:33 PM
This is fascinating! Thanks for explaining Nirosh, I feel like I understand Wi-Fi a lot better now.

Baxy Today at 9:35 PM
The timeline diagram



👍 2

Activity 2: ARP

In Cisco Packet Tracer, I'll build a network with laptops, PCs, and routers. We'll assign static IP addresses to each device and grab screenshots for reference.

First up, we need to figure out how my laptop (Laptop1) chats with PC2 on the same network. We'll discuss a protocol that helps us discover this information and the steps involved.

Then, I'll ping PC2 from my laptop in simulation mode. This will activate a cool feature called ARP. We'll analyze the messages exchanged, like what PC2 sends and the order they appear in. We'll also pay close attention to IP and MAC addresses.

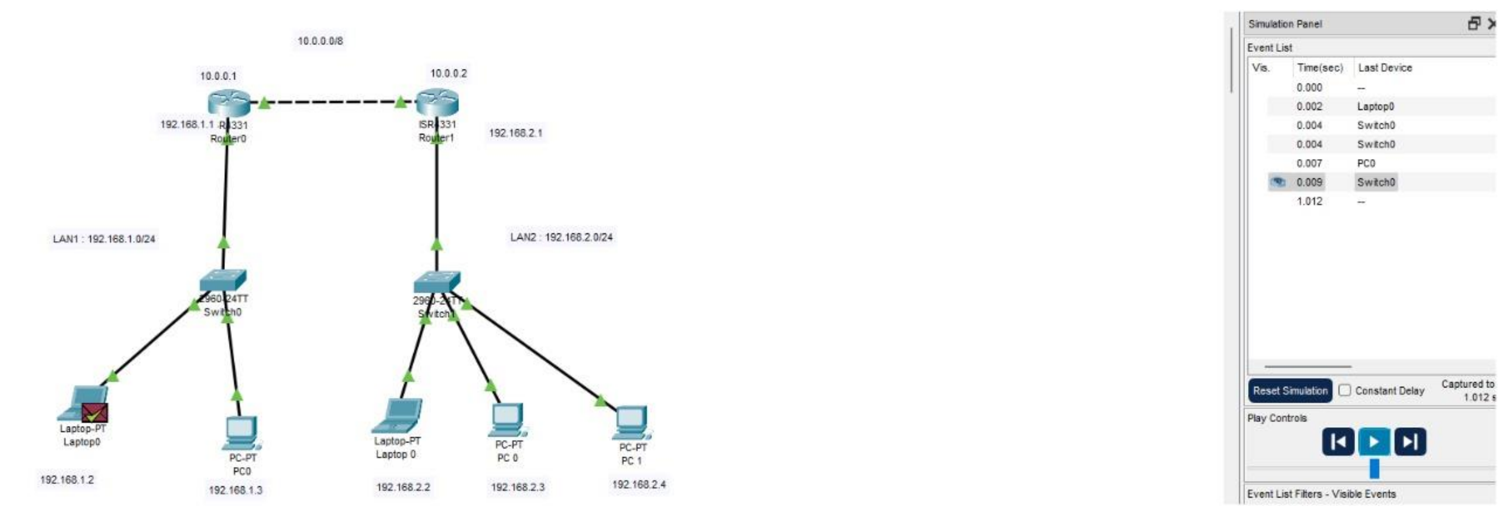
Remember how everyone keeps a cheat sheet of who's who on the network (ARP table)? We'll use commands to check these tables on various devices and see how they update as communication flows.

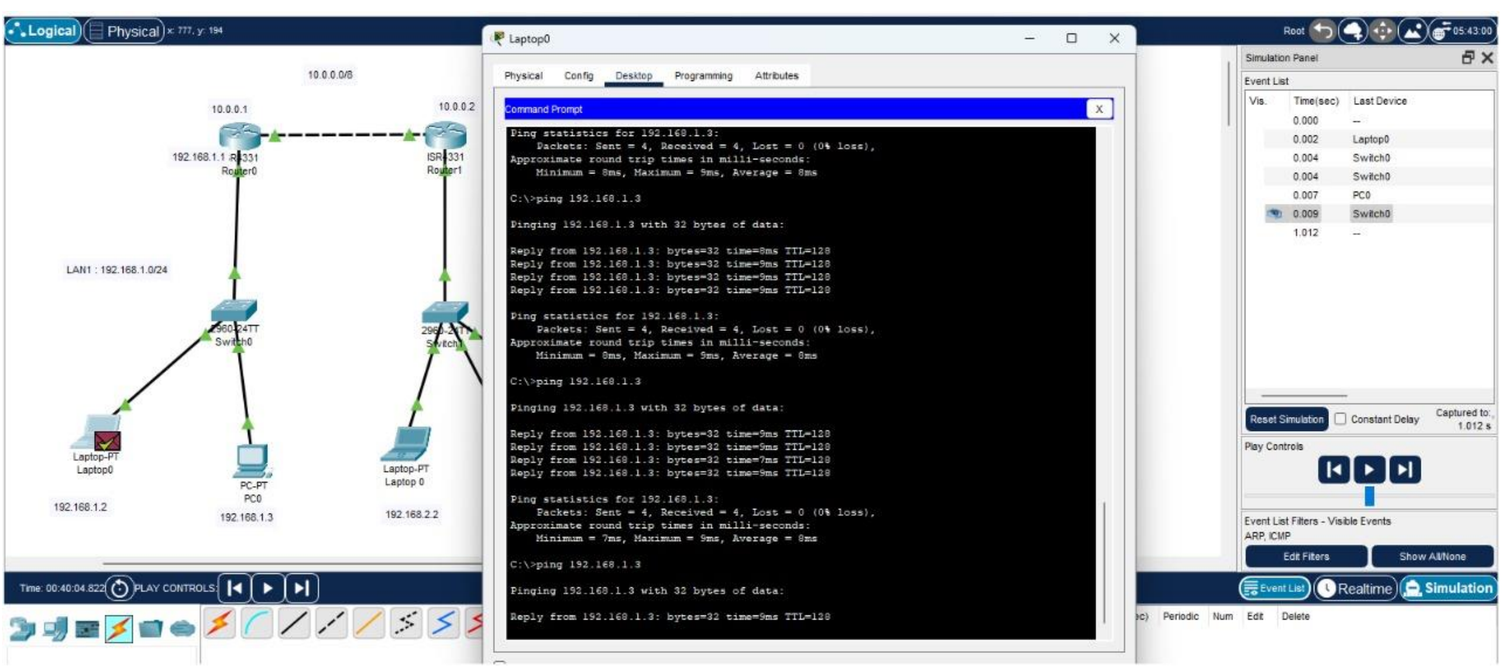
Next, I'll ping another PC (PC0) from my laptop and another laptop (Laptop0) will ping a different PC (PC1). We'll observe how the messages travel and update the cheat sheets (ARP tables) on each device.

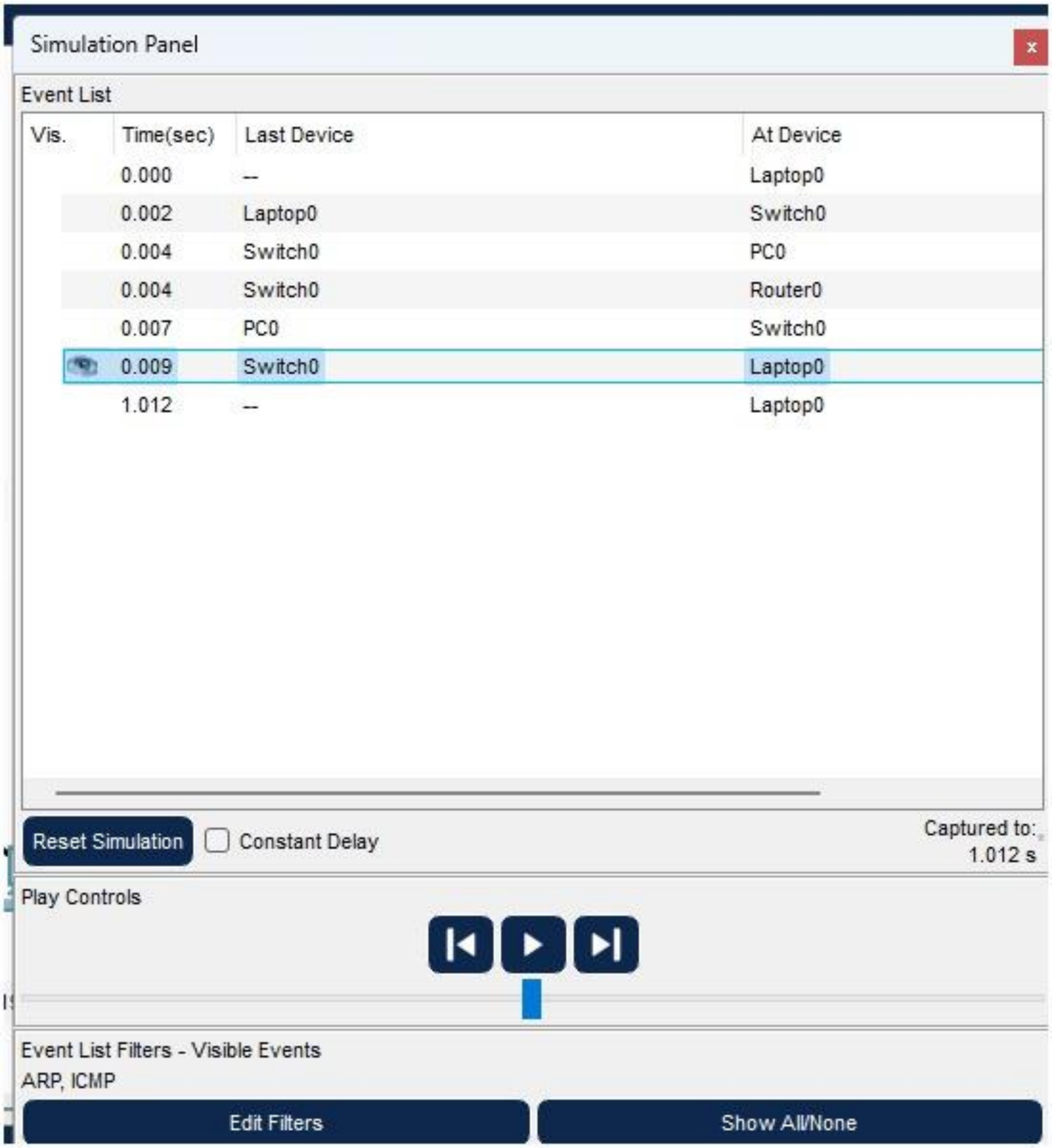
Finally, by comparing the ARP tables before and after our pings, we'll see how ARP helps devices talk to each other across the network. There's even a command to clear entries in these tables if needed!

3.

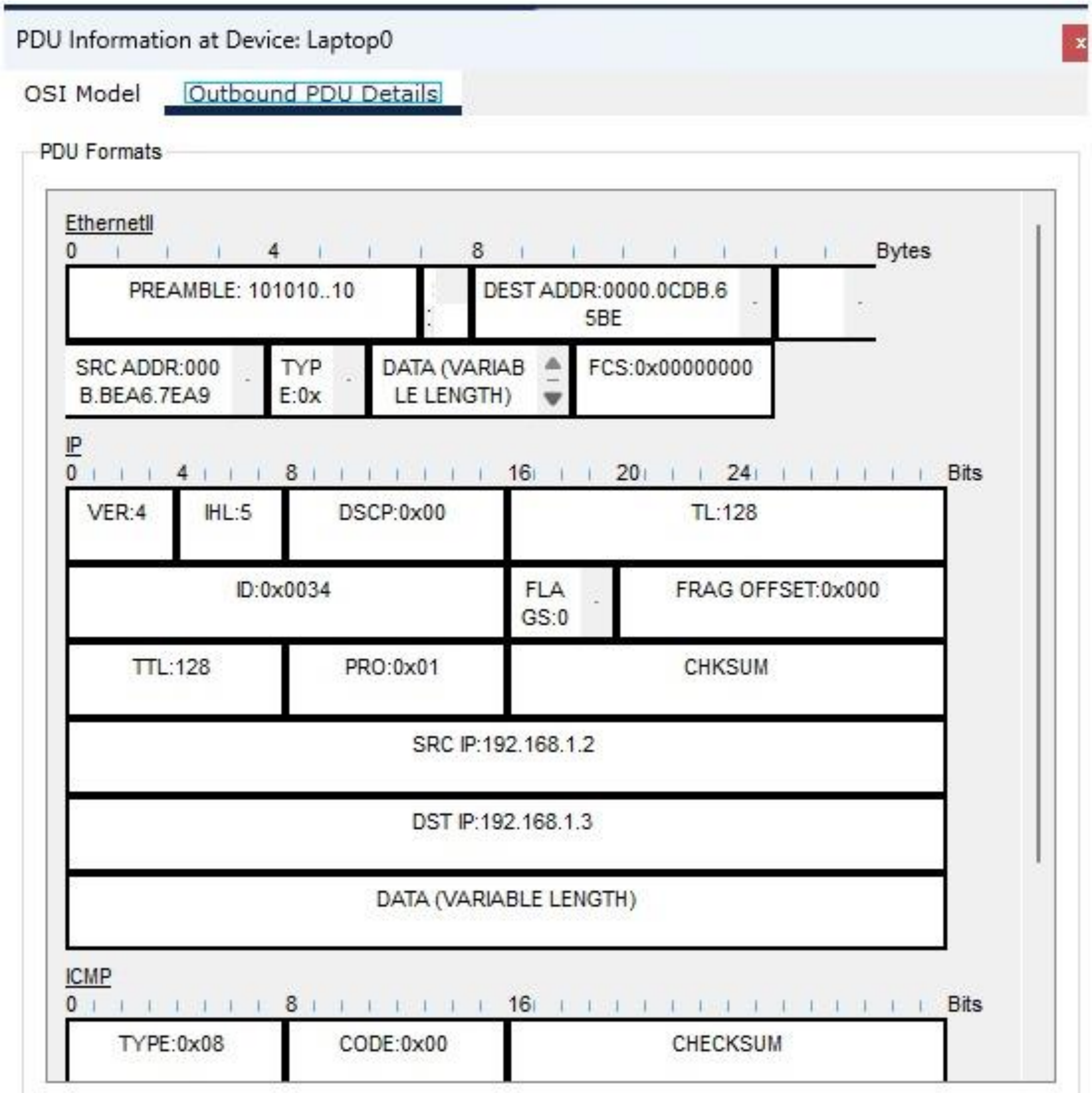
a.



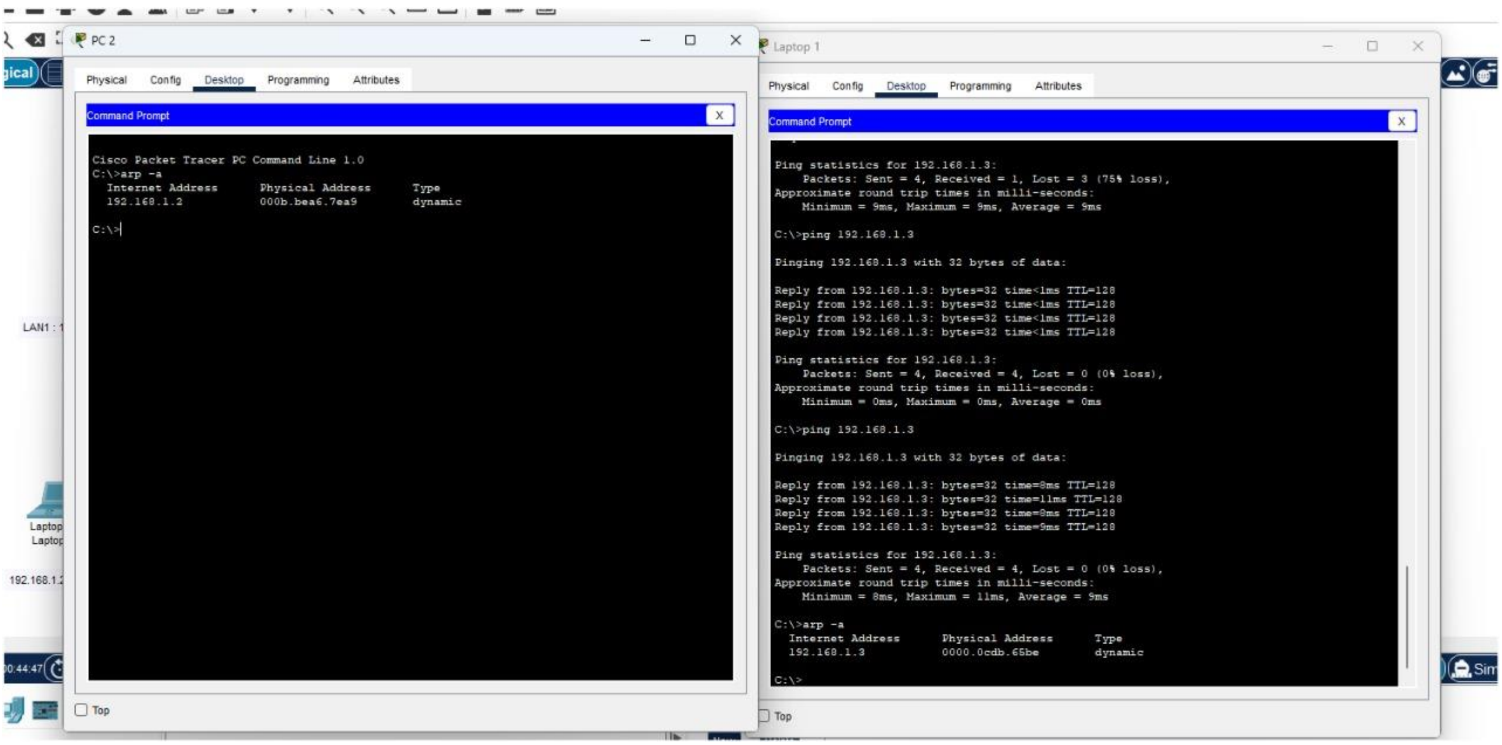



Simulation Panel
Event List
Vis. Time(sec) Last Device At Device
0.000 -- Laptop0
0.002 Laptop0 Switch0
0.004 Switch0 PC0
0.004 Switch0 Router0
0.007 PC0 Switch0
0.009 Switch0 Laptop0
1.012 -- Laptop0
Reset Simulation
Constant Delay
Captured to:
1.012 s
Play Controls
Event List Filters - Visible Events
ARP, ICMP
Edit Filters
Show All/None

b.

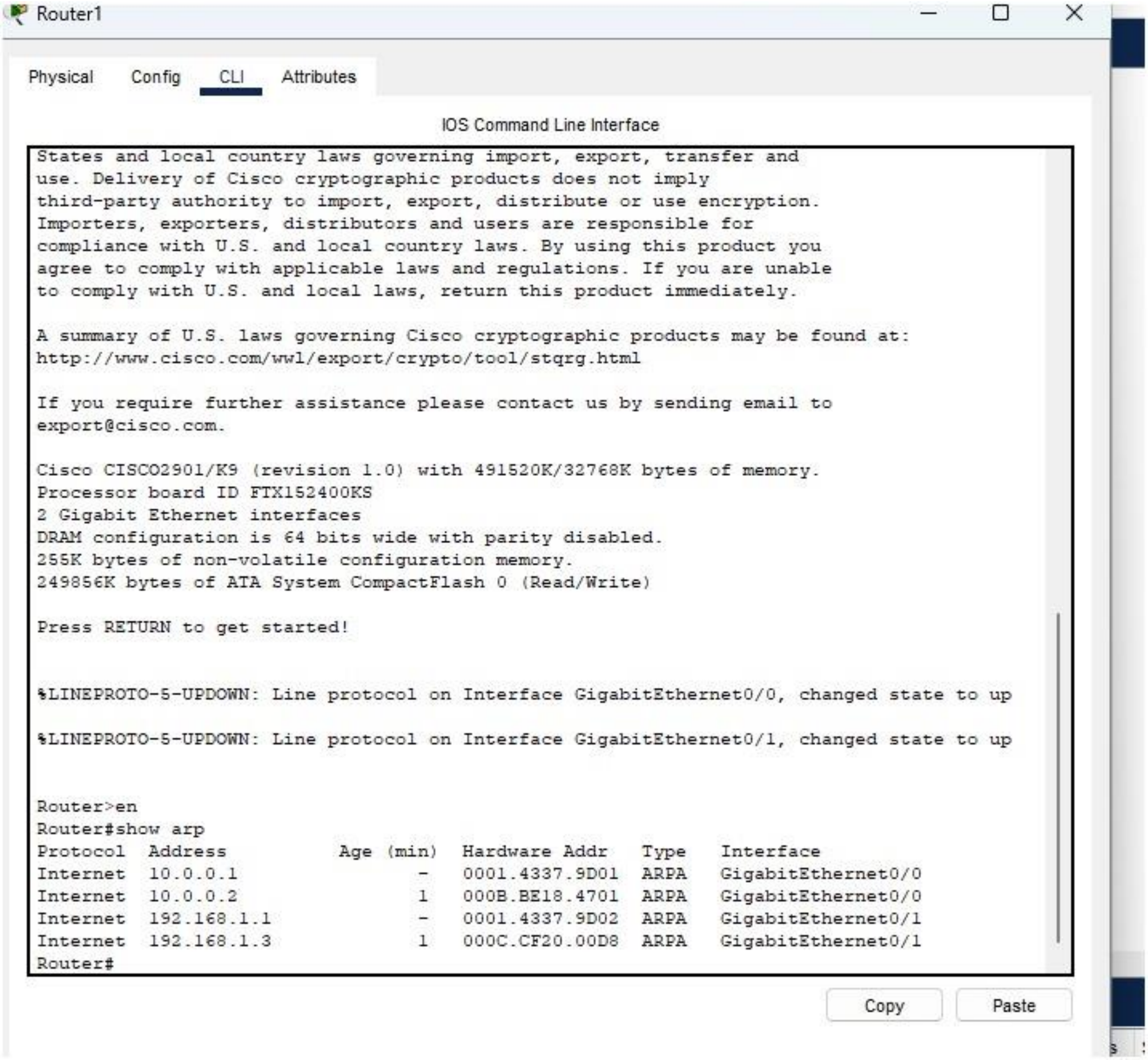

PDU Information at Device: Laptop0
OSI Model
Outbound PDU Details
PDU Formats
EthernetII
Bytes
PREAMBLE: 101010..10
DEST ADDR:0000.0CDB.65BE
SRC ADDR:000B.BEA6.7EA9
TYPE:0x
DATA (VARIABLE LENGTH)
FCS:0x00000000
IP
Bits
VER:4
IHL:5
DSCP:0x00
TL:128
ID:0x0034
FLAGS:0
FRAG OFFSET:0x000
TTL:128
PRO:0x01
CHKSUM
SRC IP:192.168.1.2
DST IP:192.168.1.3
DATA (VARIABLE LENGTH)
ICMP
Bits
TYPE:0x08
CODE:0x00
CHECKSUM

4.


PC 2
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
Internet Address Physical Address Type
192.168.1.2 000b.bea6.7ea9 dynamic
C:\>
Top
Laptop 1
Command Prompt
Ping statistics for 192.168.1.3:
Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
Approximate round trip times in milli-seconds:
Minimum = 9ms, Maximum = 9ms, Average = 9ms
C:\>ping 192.168.1.3
Pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.1.3
Pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time=8ms TTL=128
Reply from 192.168.1.3: bytes=32 time=11ms TTL=128
Reply from 192.168.1.3: bytes=32 time=8ms TTL=128
Reply from 192.168.1.3: bytes=32 time=9ms TTL=128
Ping statistics for 192.168.1.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 8ms, Maximum = 11ms, Average = 9ms
C:\>arp -a
Internet Address Physical Address Type
192.168.1.3 0000.0cdb.65be dynamic
C:\>
Top

5.

ROUTER 1


Router1
Physical
Config
CLI
Attributes
IOS Command Line Interface
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
Cisco CISCO2901/K9 (revision 1.0) with 491520K/32768K bytes of memory.
Processor board ID FTX152400KS
2 Gigabit Ethernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)
Press RETURN to get started!
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
Router>en
Router#show arp
Protocol Address   Age (min) Hardware Addr   Type   Interface
Internet 10.0.0.1   -   0001.4337.9D01 ARPA   GigabitEthernet0/0
Internet 10.0.0.2   1   000B.BE18.4701 ARPA   GigabitEthernet0/0
Internet 192.168.1.1   -   0001.4337.9D02 ARPA   GigabitEthernet0/1
Internet 192.168.1.3   1   000C.CF20.00D8 ARPA   GigabitEthernet0/1
Router#
Copy
Paste

ROUTER 3

Router3

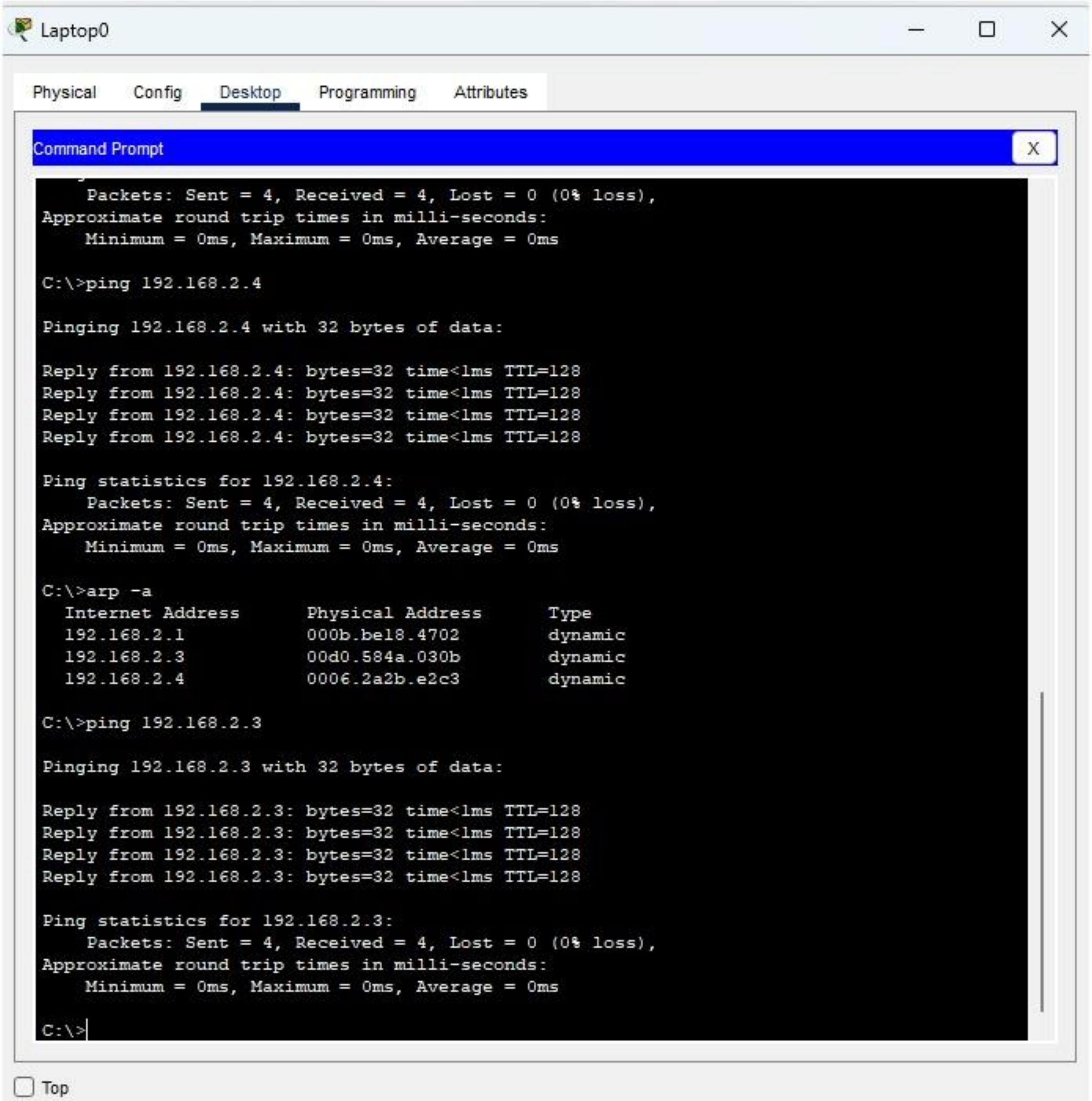
Physical Config CLI Attributes

IOS Command Line Interface

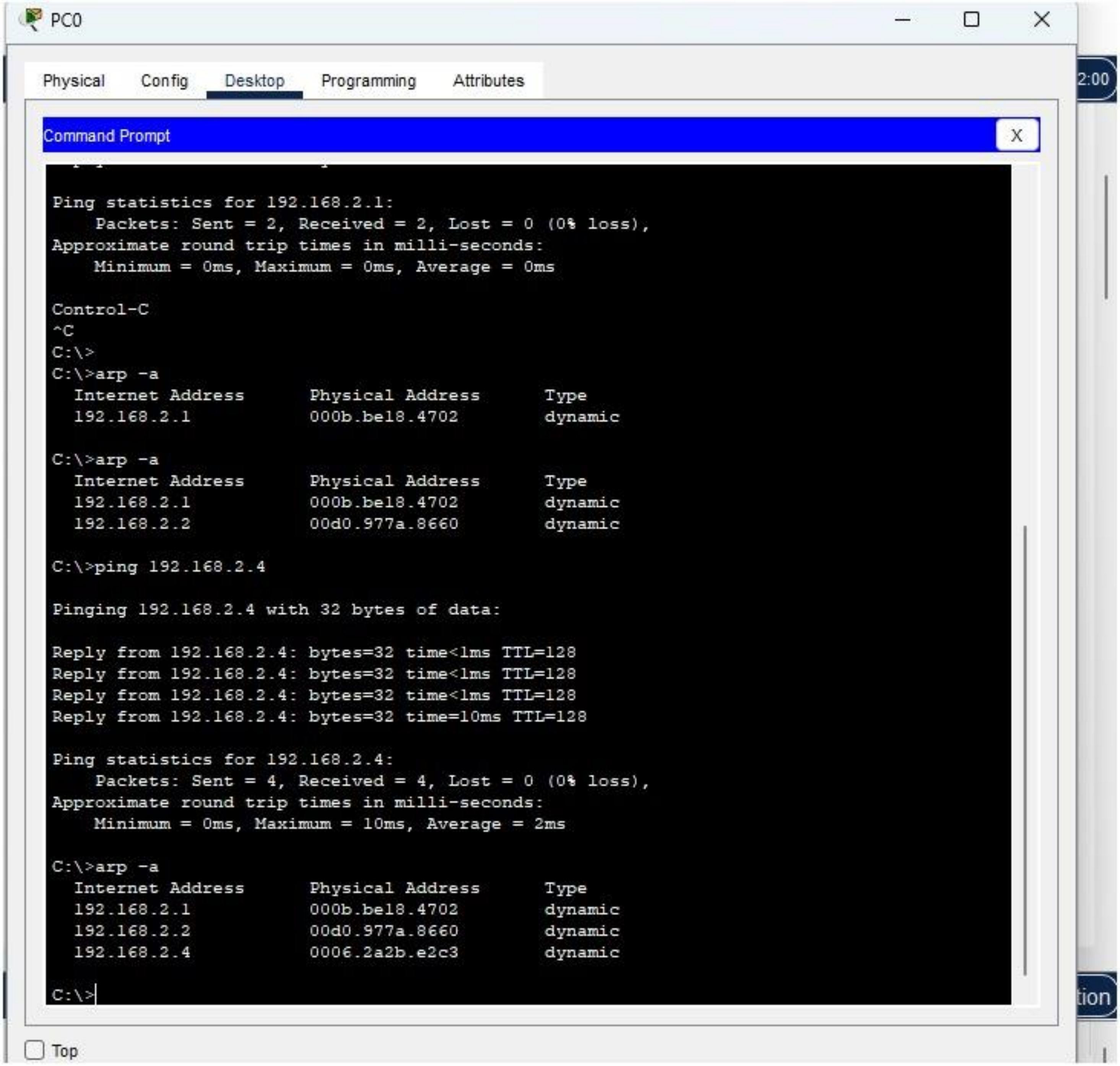
```
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

Cisco CISCO2901/K9 (revision 1.0) with 491520K/32768K bytes of memory.
Processor board ID FTX152400KS
2 Gigabit Ethernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)

Press RETURN to get started!


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

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


Router>show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.0.0.1                1   0001.4337.9D01  ARPA   GigabitEthernet0/0
Internet  10.0.0.2                -   000B.BE18.4701  ARPA   GigabitEthernet0/0
Internet  192.168.2.1             -   000B.BE18.4702  ARPA   GigabitEthernet0/1
Internet  192.168.2.3             1   00D0.584A.030B  ARPA   GigabitEthernet0/1
Internet  192.168.2.4             1   0006.2A2B.E2C3  ARPA   GigabitEthernet0/1
Router>
```

Copy Paste

LAPTOP 0


Laptop0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.2.4
Pinging 192.168.2.4 with 32 bytes of data:
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>arp -a
  Internet Address   Physical Address   Type
  192.168.2.1   000b.be18.4702   dynamic
  192.168.2.3   00d0.584a.030b   dynamic
  192.168.2.4   0006.2a2b.e2c3   dynamic
C:\>ping 192.168.2.3
Pinging 192.168.2.3 with 32 bytes of data:
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
Top

PC 0


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Ping statistics for 192.168.2.1:
Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\>
C:\>arp -a
Internet Address Physical Address Type
192.168.2.1 000b.be18.4702 dynamic
C:\>arp -a
Internet Address Physical Address Type
192.168.2.1 000b.be18.4702 dynamic
192.168.2.2 00d0.977a.8660 dynamic
C:\>ping 192.168.2.4
Pinging 192.168.2.4 with 32 bytes of data:
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Reply from 192.168.2.4: bytes=32 time=10ms TTL=128
Ping statistics for 192.168.2.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 10ms, Average = 2ms
C:\>arp -a
Internet Address Physical Address Type
192.168.2.1 000b.be18.4702 dynamic
192.168.2.2 00d0.977a.8660 dynamic
192.168.2.4 0006.2a2b.e2c3 dynamic
C:\>
Top

PC 1

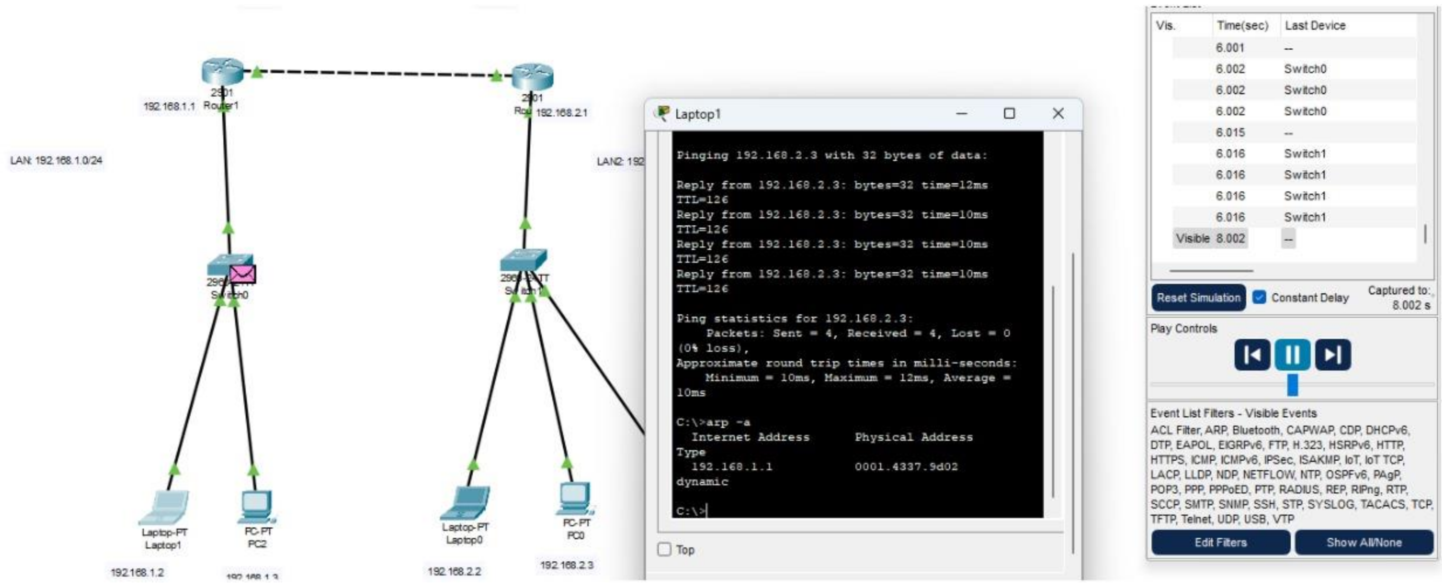
Physical Config Desktop Programming Attributes

Command Prompt

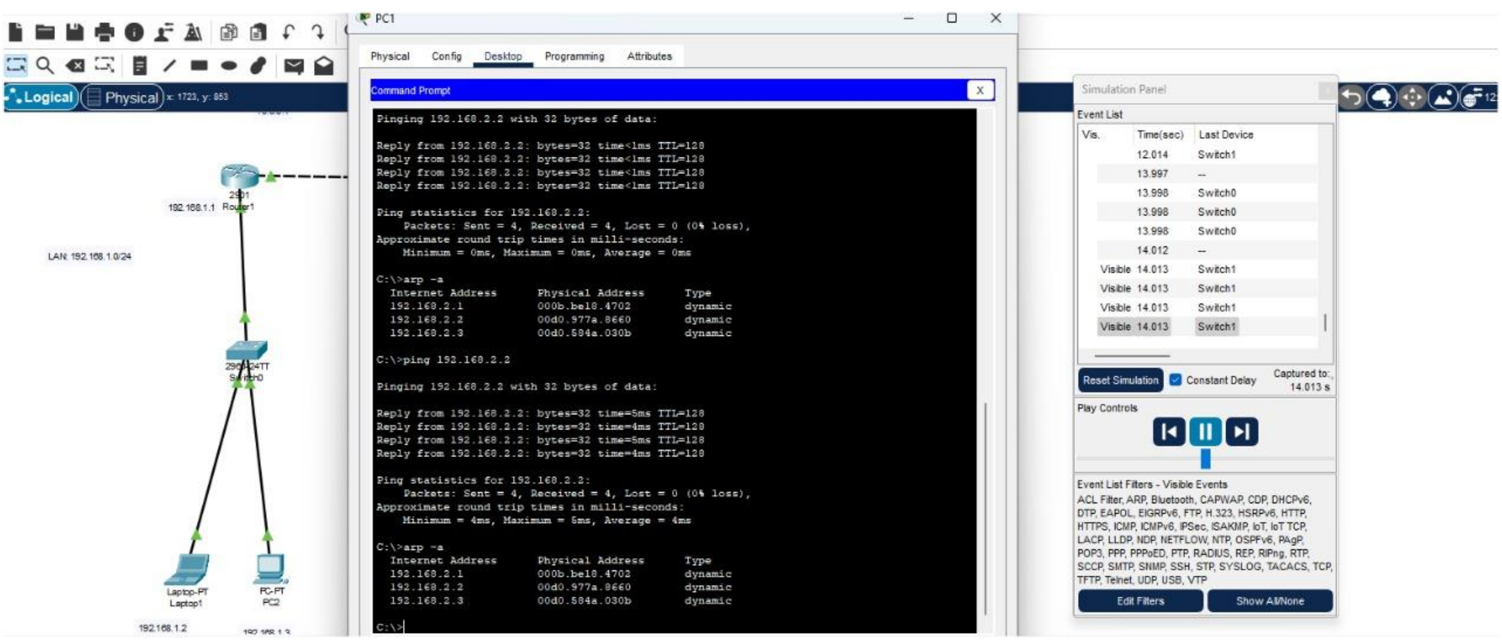
```
C:\>arp -a
  Internet Address      Physical Address      Type
  192.168.2.1           000b.be18.4702        dynamic

C:\>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time<1ms TTL=128
Reply from 192.168.2.3: bytes=32 time=1ms TTL=128
Reply from 192.168.2.3: bytes=32 time=5ms TTL=128
Reply from 192.168.2.3: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 5ms, Average = 1ms

C:\>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>arp -a
  Internet Address      Physical Address      Type
  192.168.2.1           000b.be18.4702        dynamic
  192.168.2.2           00d0.977a.8660        dynamic
  192.168.2.3           00d0.584a.030b        dynamic

C:\>
```

6.

LAPTOP 1 TO PC 0

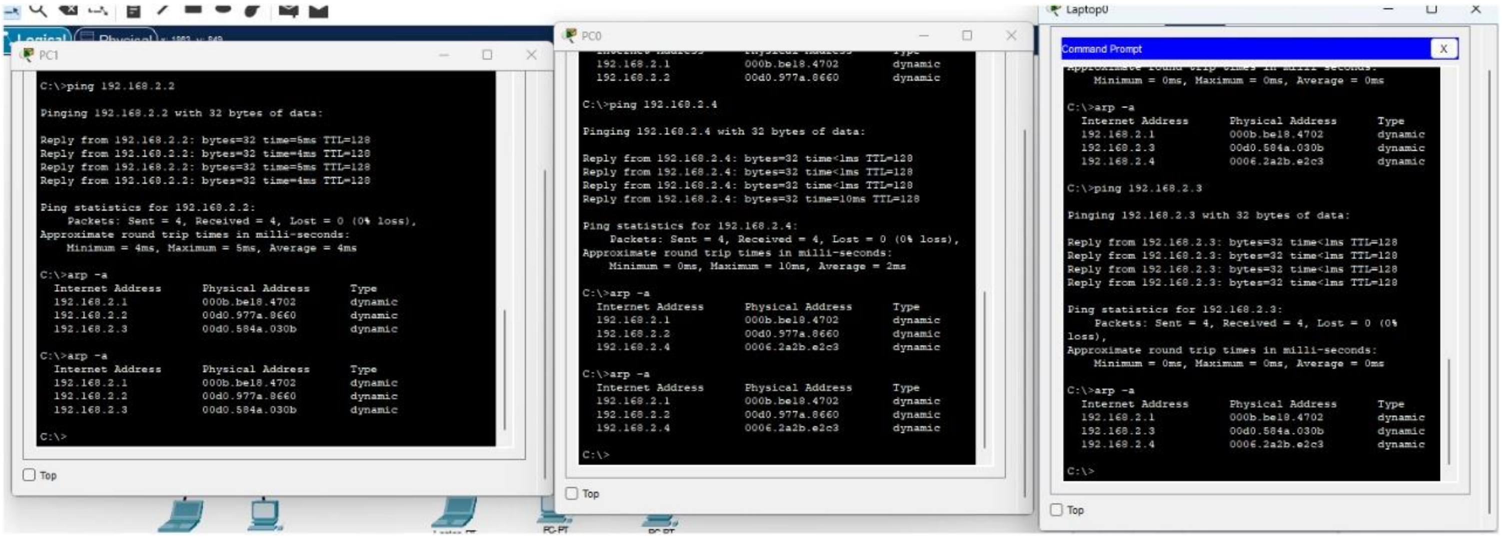


PC 1 TO LAPTOP 0

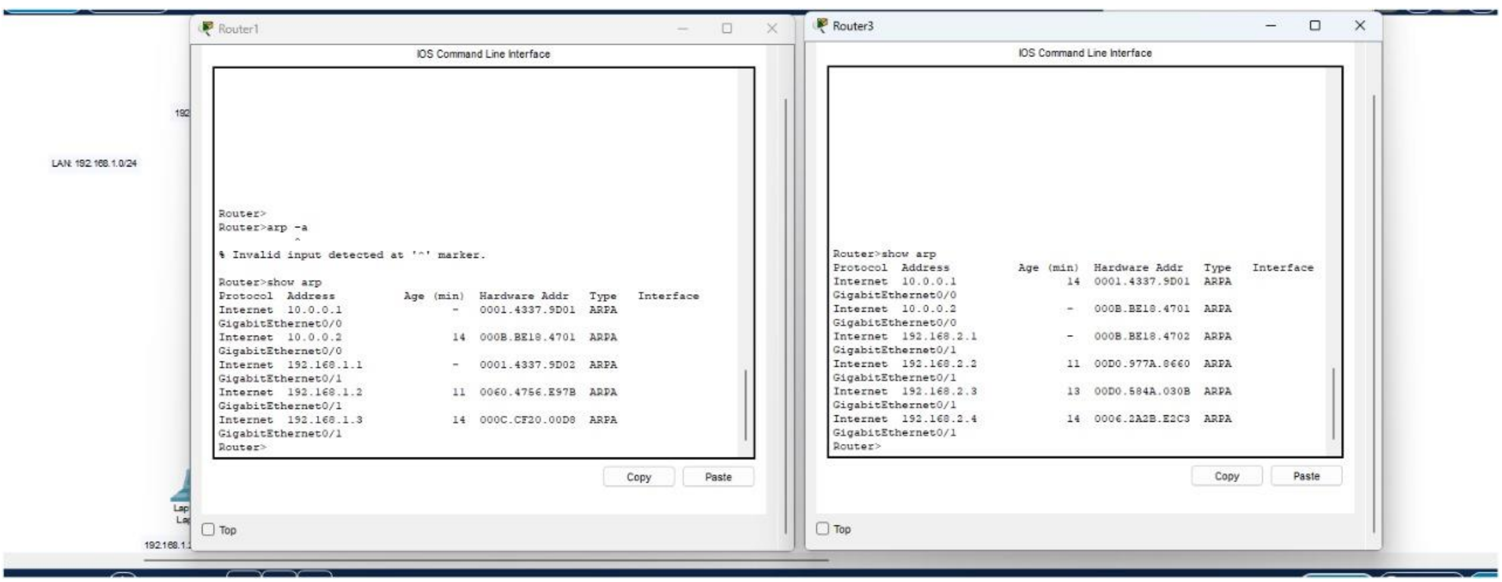


7.

LAPTOP 0, PC 0, PC1



ROUTER 1 AND ROUTER 3



Activity 3: Mapping physical connections – MAC address table.

In Cisco Packet Tracer, I'll build a network with PCs, laptops, and a switch. First, I'll write down the MAC addresses of all the devices to get to know them better. We'll also check a list each PC and laptop keeps to remember other devices on the network (ARP table). The switch has its own list too, called a MAC address table, and we'll see what's in there using a special command.

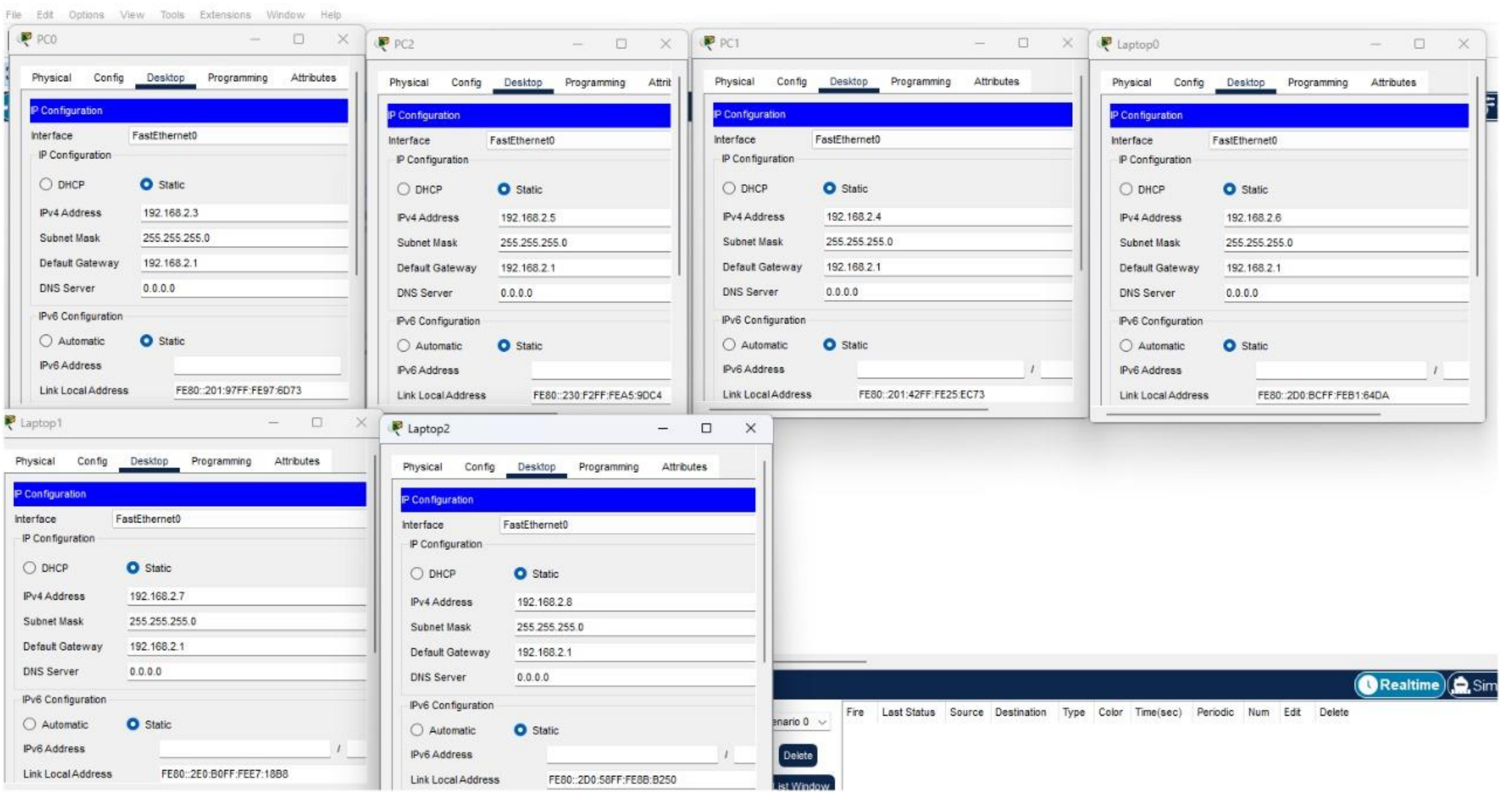
Then, it's communication time!

I'll ping Laptops 0 and 1 from different PCs. This will update the switch's list, and we'll watch it change with a cool magnifying glass tool. We'll even ping another laptop to see how the switch learns where devices are. To see the bigger picture, we'll check the ARP tables on the PCs and laptops again.

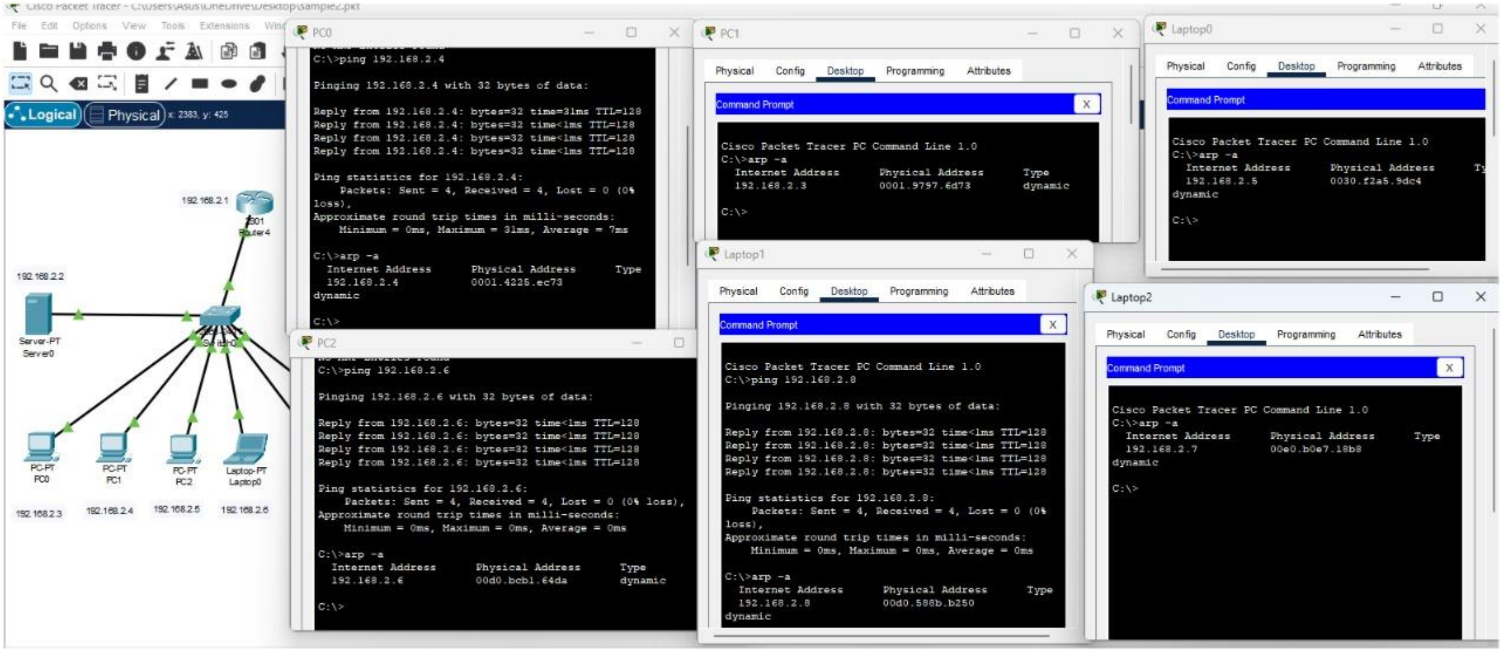
Next, we'll erase the switch's memory of all the devices. But instead of manually assigning IP addresses, we'll set up a DHCP server to do it automatically. We'll compare the new lists on the PCs and laptops (ARP tables) to what we saw before. Finally, we'll peek at the switch's list again and see how it differs. This will help us understand the difference between using a DHCP server and static IP assignment.

1.

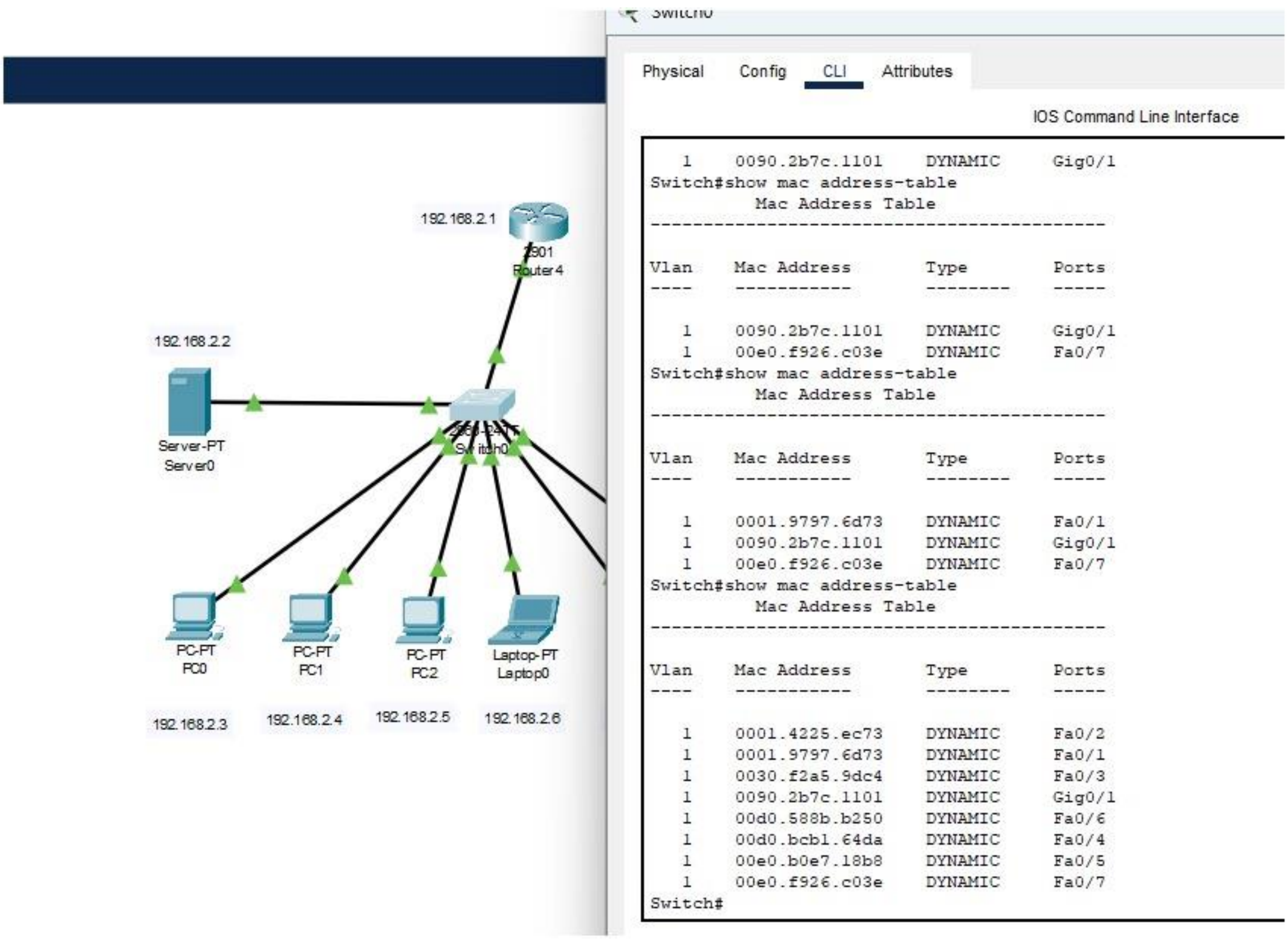
a.



b.

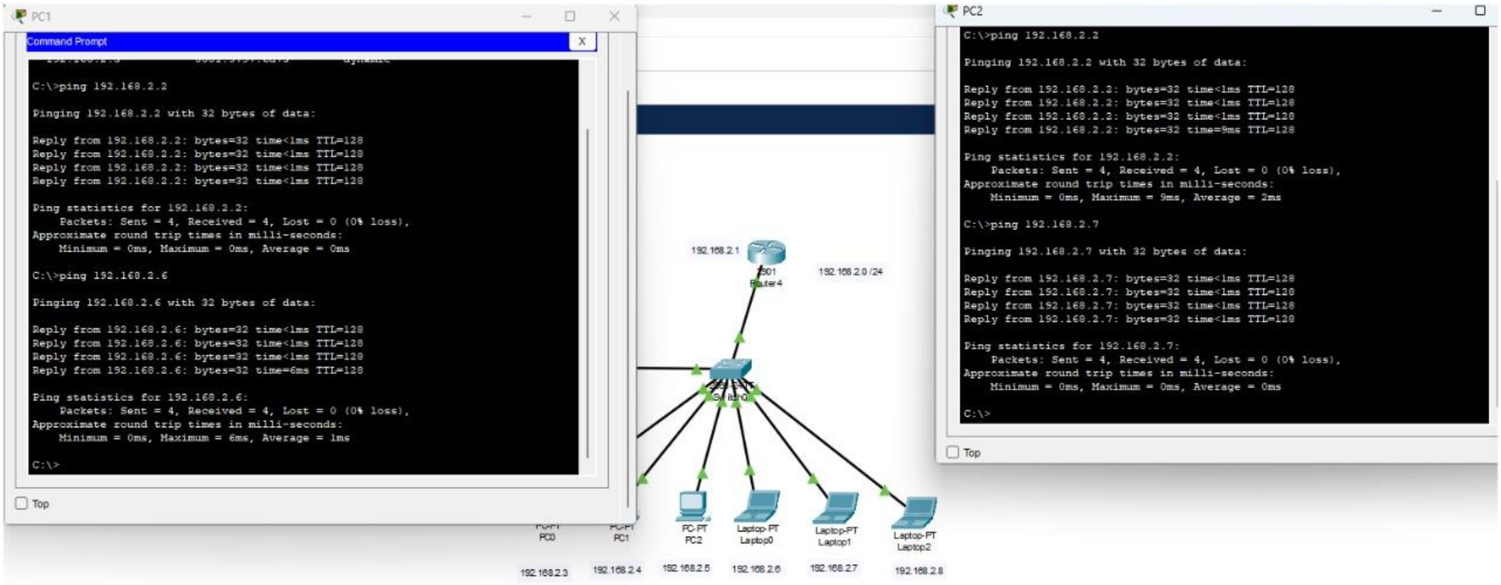

PC0
C:\>ping 192.168.2.4
Pinging 192.168.2.4 with 32 bytes of data:
Reply from 192.168.2.4: bytes=32 time=31ms TTL=128
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Reply from 192.168.2.4: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 31ms, Average = 7ms
C:\>arp -a
Internet Address Physical Address Type
192.168.2.4 0001.4225.ec73 dynamic
C:\>
PC1
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
Internet Address Physical Address Type
192.168.2.3 0001.9797.6d73 dynamic
C:\>
Laptop0
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
Internet Address Physical Address
192.168.2.5 0030.f2a5.9dc4
dynamic
C:\>
PC2
C:\>ping 192.168.2.6
Pinging 192.168.2.6 with 32 bytes of data:
Reply from 192.168.2.6: bytes=32 time<1ms TTL=128
Reply from 192.168.2.6: bytes=32 time<1ms TTL=128
Reply from 192.168.2.6: bytes=32 time<1ms TTL=128
Reply from 192.168.2.6: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.6:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>arp -a
Internet Address Physical Address Type
192.168.2.6 00d0.bcb1.64da dynamic
C:\>
Laptop1
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.2.8
Pinging 192.168.2.8 with 32 bytes of data:
Reply from 192.168.2.8: bytes=32 time<1ms TTL=128
Reply from 192.168.2.8: bytes=32 time<1ms TTL=128
Reply from 192.168.2.8: bytes=32 time<1ms TTL=128
Reply from 192.168.2.8: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.8:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>arp -a
Internet Address Physical Address Type
192.168.2.8 00d0.588b.b250 dynamic
Laptop2
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
Internet Address Physical Address Type
192.168.2.7 00e0.b0e7.18b8 dynamic
C:\>

c.

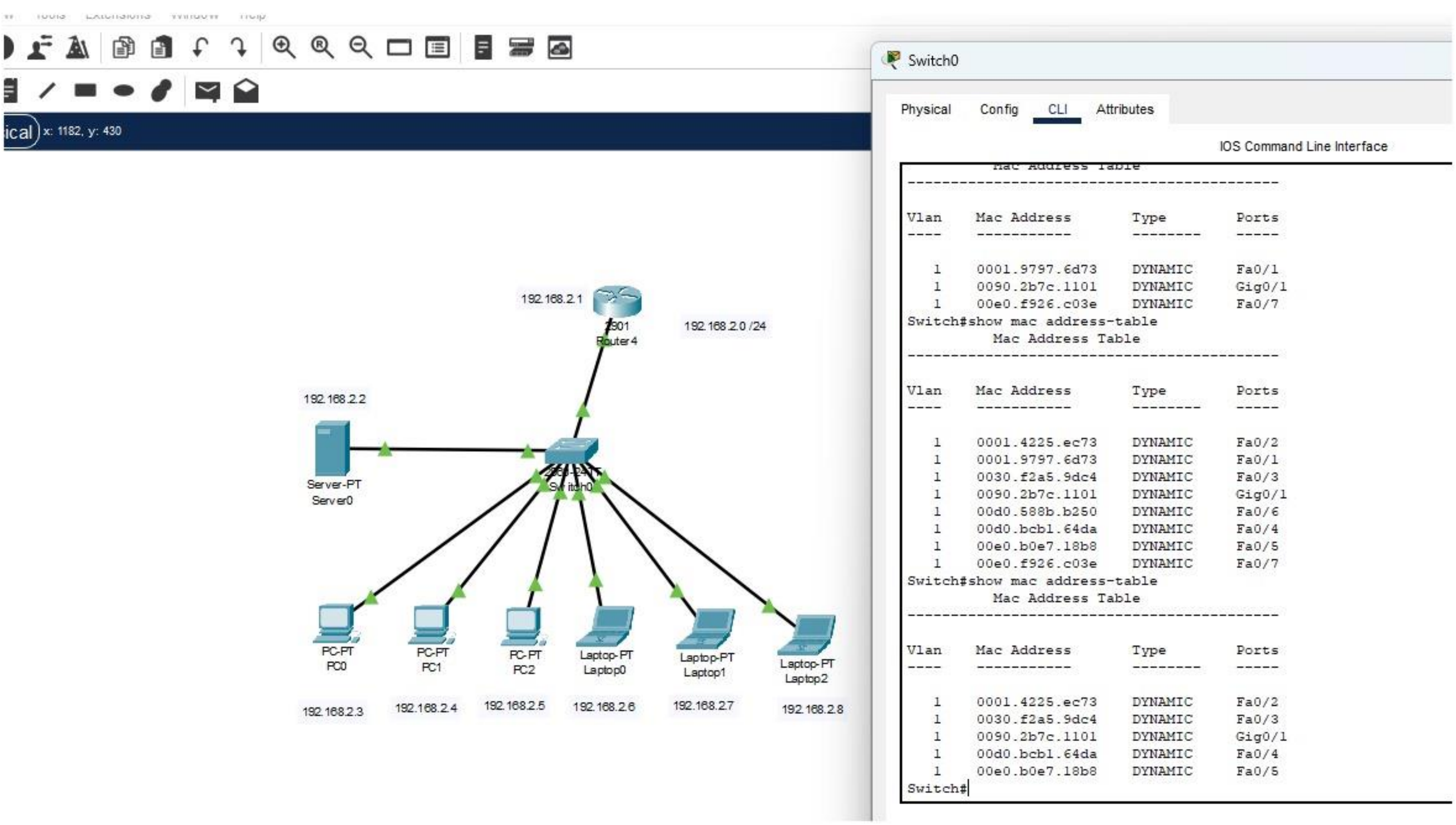

Physical
Config
CLI
Attributes
IOS Command Line Interface
1 0090.2b7c.1101 DYNAMIC Gig0/1
Switch#show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00e0.f926.c03e DYNAMIC Fa0/7
Switch#show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0001.9797.6d73 DYNAMIC Fa0/1
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00e0.f926.c03e DYNAMIC Fa0/7
Switch#show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0001.4225.ec73 DYNAMIC Fa0/2
1 0001.9797.6d73 DYNAMIC Fa0/1
1 0030.f2a5.9dc4 DYNAMIC Fa0/3
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00d0.588b.b250 DYNAMIC Fa0/6
1 00d0.bcb1.64da DYNAMIC Fa0/4
1 00e0.b0e7.18b8 DYNAMIC Fa0/5
1 00e0.f926.c03e DYNAMIC Fa0/7
Switch#
192.168.2.1
2901
Router4
192.168.2.2
Server-PT
Server0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Laptop-PT
Laptop0
192.168.2.3
192.168.2.4
192.168.2.5
192.168.2.6

d.

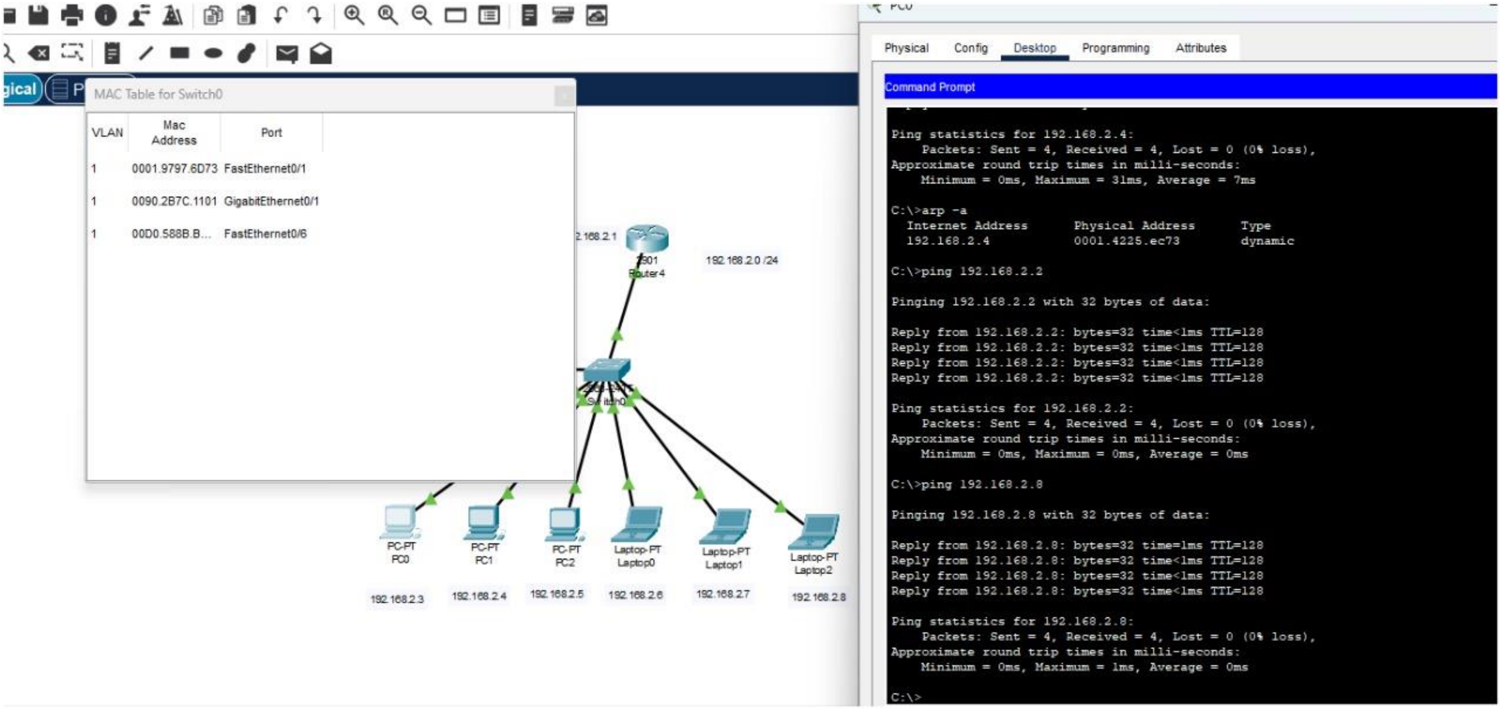
e.


PC1
Command Prompt
C:\>ping 192.168.2.2
Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.2.6
Pinging 192.168.2.6 with 32 bytes of data:
Reply from 192.168.2.6: bytes=32 time<1ms TTL=128
Reply from 192.168.2.6: bytes=32 time<1ms TTL=128
Reply from 192.168.2.6: bytes=32 time<1ms TTL=128
Reply from 192.168.2.6: bytes=32 time=6ms TTL=128
Ping statistics for 192.168.2.6:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 6ms, Average = 1ms
C:\>
Top
PC2
C:\>ping 192.168.2.2
Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time=9ms TTL=128
Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 9ms, Average = 2ms
C:\>ping 192.168.2.7
Pinging 192.168.2.7 with 32 bytes of data:
Reply from 192.168.2.7: bytes=32 time<1ms TTL=128
Reply from 192.168.2.7: bytes=32 time<1ms TTL=128
Reply from 192.168.2.7: bytes=32 time<1ms TTL=128
Reply from 192.168.2.7: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.7:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
Top
192.168.2.1
192.168.2.0 /24
PC-PT PC2
Laptop-PT Laptop0
Laptop-PT Laptop1
Laptop-PT Laptop2
192.168.2.3
192.168.2.4
192.168.2.5
192.168.2.6
192.168.2.7
192.168.2.8

f.

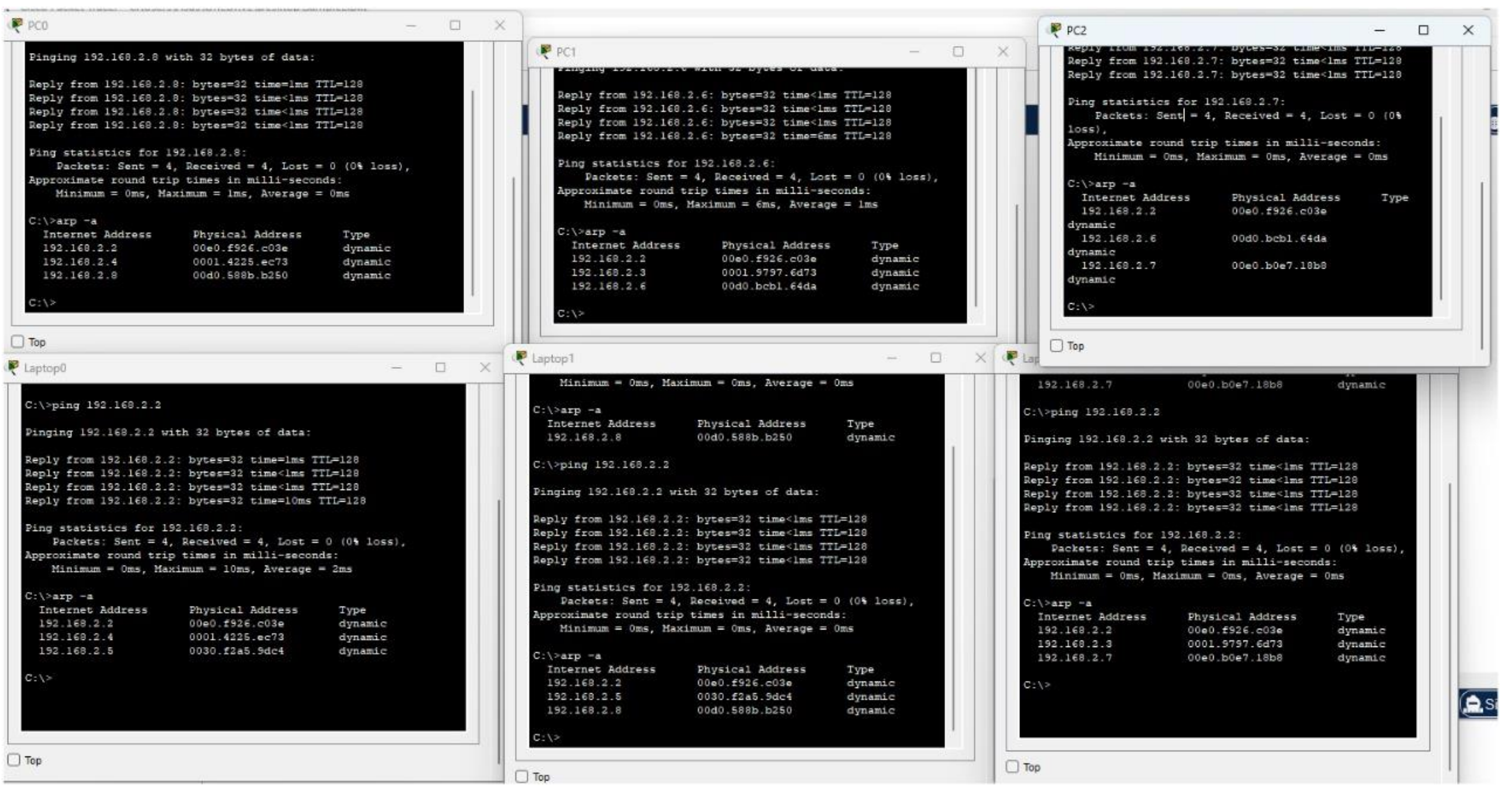

x: 1182, y: 430
Switch0
Physical
Config
CLI
Attributes
IOS Command Line Interface
Vlan Mac Address Type Ports
1 0001.9797.6d73 DYNAMIC Fa0/1
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00e0.f926.c03e DYNAMIC Fa0/7
Switch#show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0001.4225.ec73 DYNAMIC Fa0/2
1 0001.9797.6d73 DYNAMIC Fa0/1
1 0030.f2a5.9dc4 DYNAMIC Fa0/3
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00d0.588b.b250 DYNAMIC Fa0/6
1 00d0.bcb1.64da DYNAMIC Fa0/4
1 00e0.b0e7.18b8 DYNAMIC Fa0/5
1 00e0.f926.c03e DYNAMIC Fa0/7
Switch#show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0001.4225.ec73 DYNAMIC Fa0/2
1 0030.f2a5.9dc4 DYNAMIC Fa0/3
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00d0.bcb1.64da DYNAMIC Fa0/4
1 00e0.b0e7.18b8 DYNAMIC Fa0/5
Switch#
192.168.2.1
Router4
192.168.2.0 /24
192.168.2.2
Server-PT
Server0
Switch0
PC-PT PC0
PC-PT PC1
PC-PT PC2
Laptop-PT Laptop0
Laptop-PT Laptop1
Laptop-PT Laptop2
192.168.2.3
192.168.2.4
192.168.2.5
192.168.2.6
192.168.2.7
192.168.2.8

g.h.



MAC Table for Switch0
VLAN Mac Address Port
1 0001.9797.6D73 FastEthernet0/1
1 0090.2B7C.1101 GigabitEthernet0/1
1 00D0.588B.B... FastEthernet0/6
192.168.2.0 /24
PC-PT PC0
PC-PT PC1
PC-PT PC2
Laptop-PT Laptop0
Laptop-PT Laptop1
Laptop-PT Laptop2
192.168.2.3
192.168.2.4
192.168.2.5
192.168.2.6
192.168.2.7
192.168.2.8
Physical Config Desktop Programming Attributes
Command Prompt
Ping statistics for 192.168.2.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 31ms, Average = 7ms
C:\>arp -a
Internet Address Physical Address Type
192.168.2.4 0001.4225.ec73 dynamic
C:\>ping 192.168.2.2
Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Reply from 192.168.2.2: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.2.8
Pinging 192.168.2.8 with 32 bytes of data:
Reply from 192.168.2.8: bytes=32 time=1ms TTL=128
Reply from 192.168.2.8: bytes=32 time<1ms TTL=128
Reply from 192.168.2.8: bytes=32 time<1ms TTL=128
Reply from 192.168.2.8: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.2.8:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>

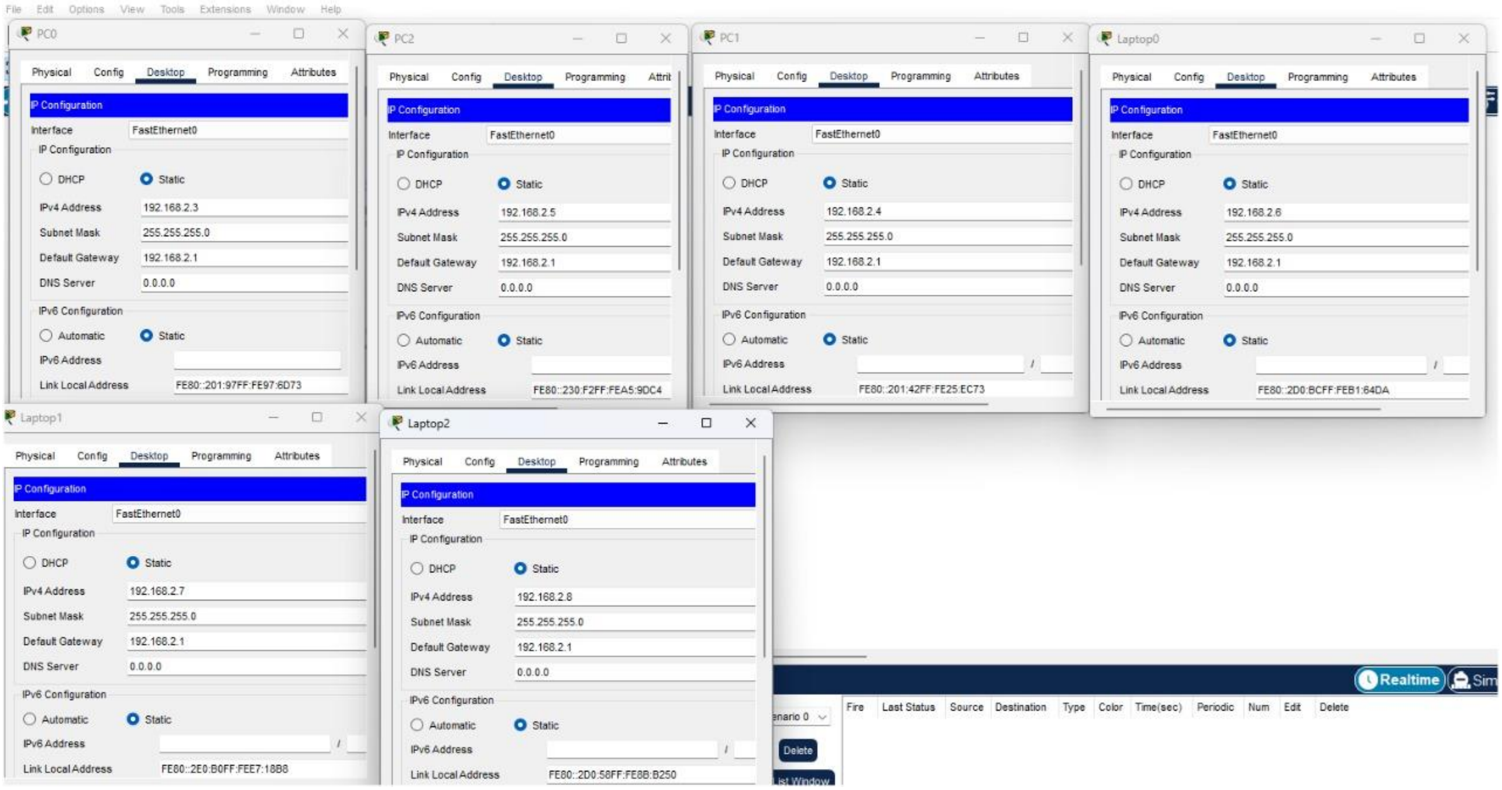
i.



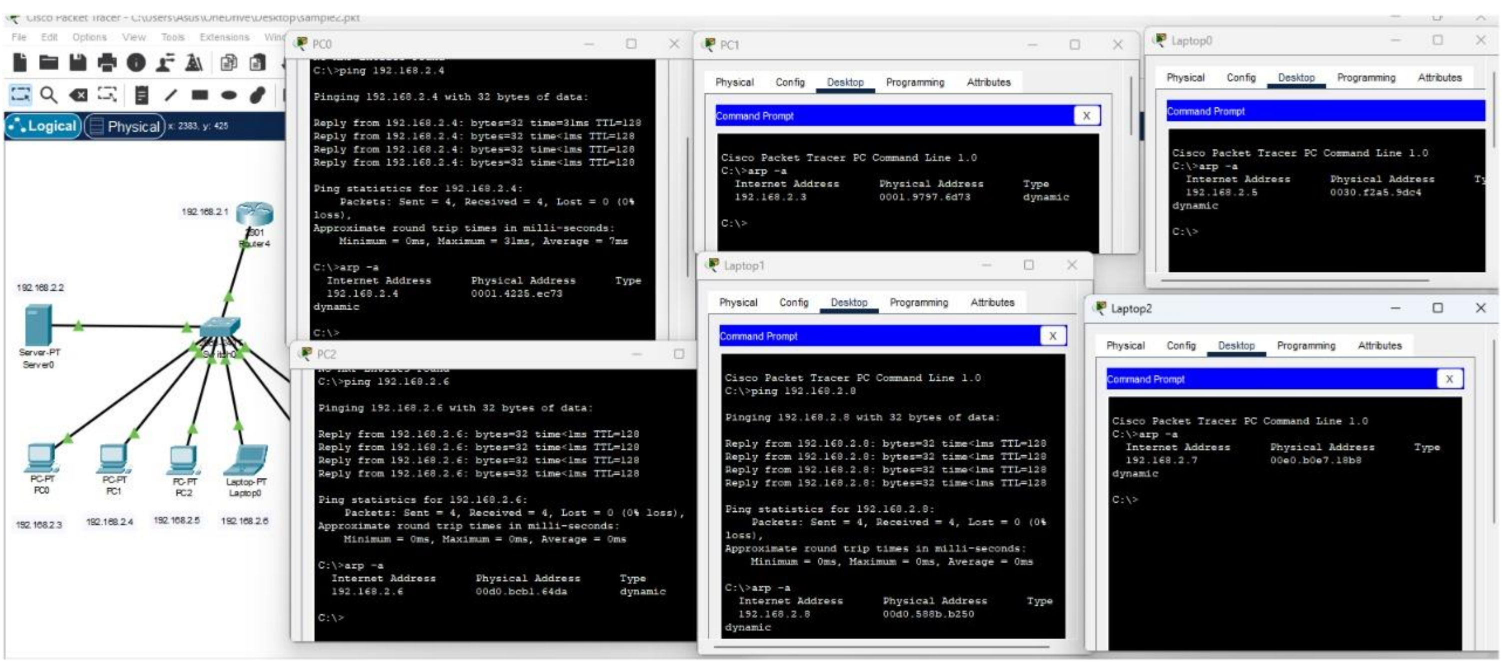
PC0
PC1
PC2
Laptop0
Laptop1

2.

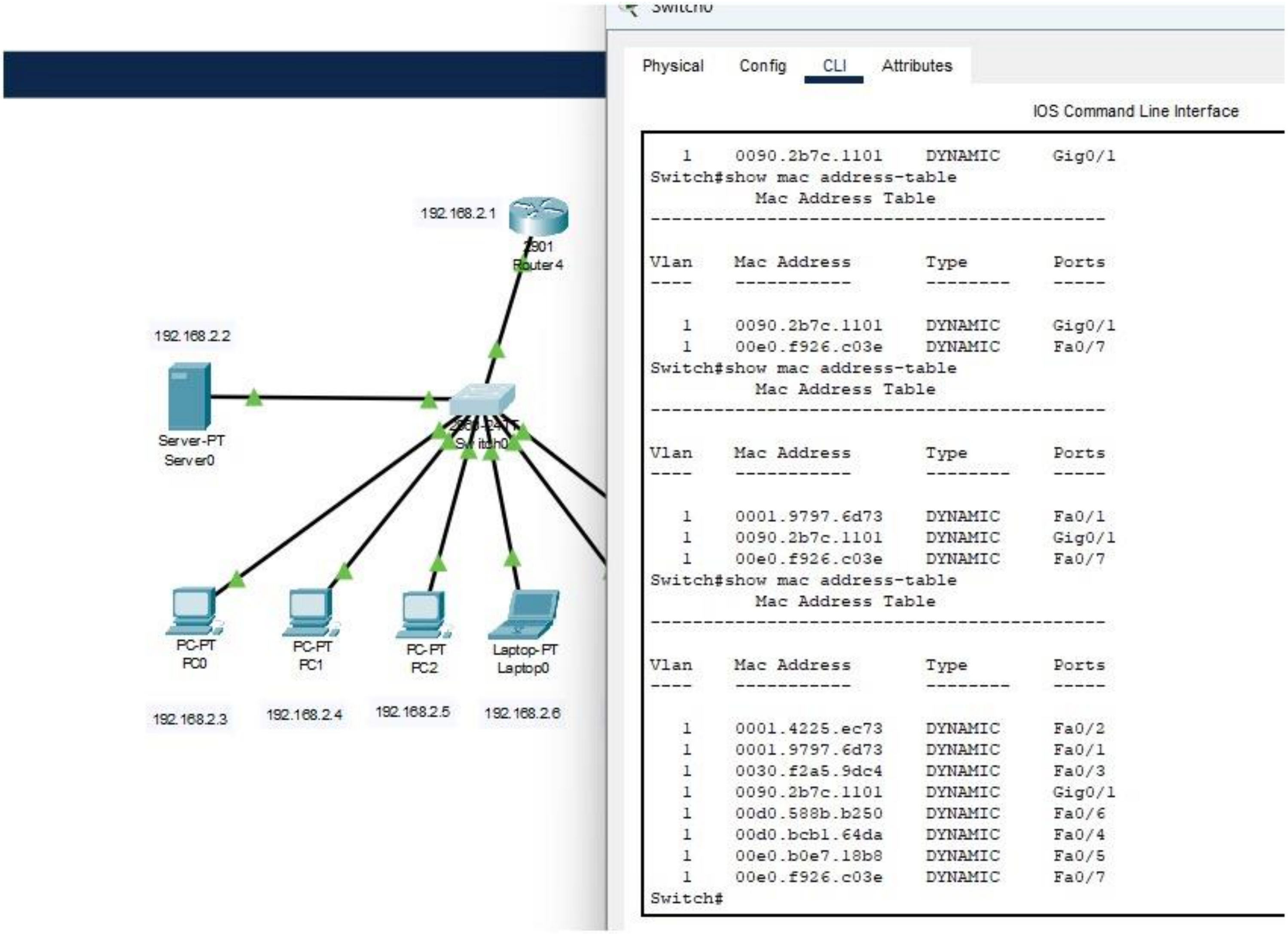
a.



b.



c.



Switch0
Physical
Config
CLI
Attributes
IOS Command Line Interface
1 0090.2b7c.1101 DYNAMIC Gig0/1
Switch#show mac address-table
Mac Address Table

Vlan Mac Address Type Ports
---- ----------- -------- -----
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00e0.f926.c03e DYNAMIC Fa0/7
Switch#show mac address-table
Mac Address Table

Vlan Mac Address Type Ports
---- ----------- -------- -----
1 0001.9797.6d73 DYNAMIC Fa0/1
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00e0.f926.c03e DYNAMIC Fa0/7
Switch#show mac address-table
Mac Address Table

Vlan Mac Address Type Ports
---- ----------- -------- -----
1 0001.4225.ec73 DYNAMIC Fa0/2
1 0001.9797.6d73 DYNAMIC Fa0/1
1 0030.f2a5.9dc4 DYNAMIC Fa0/3
1 0090.2b7c.1101 DYNAMIC Gig0/1
1 00d0.588b.b250 DYNAMIC Fa0/6
1 00d0.bcb1.64da DYNAMIC Fa0/4
1 00e0.b0e7.18b8 DYNAMIC Fa0/5
1 00e0.f926.c03e DYNAMIC Fa0/7
Switch#
192.168.2.1
Router4
192.168.2.2
Server-PT
Server0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Laptop-PT
Laptop0
192.168.2.3
192.168.2.4
192.168.2.5
192.168.2.6