**IT 5423 – Lab 10**

**Total Points: 100**

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objectives**

This lab includes the following tasks:

* Perform ARP and understand how ARP works when routers are involved

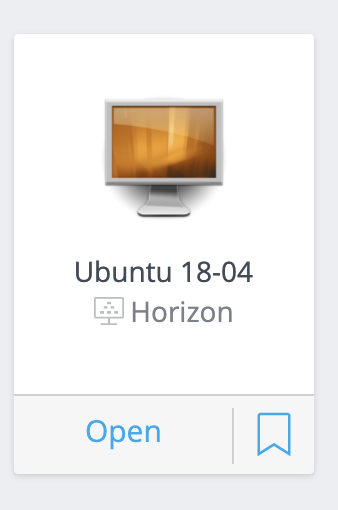
**Due Date and Submission Procedure**

* Due Date: Sunday, November 20th, 11:30 pm
* Submit your report to D2L in the assignment Assignment5-Submission

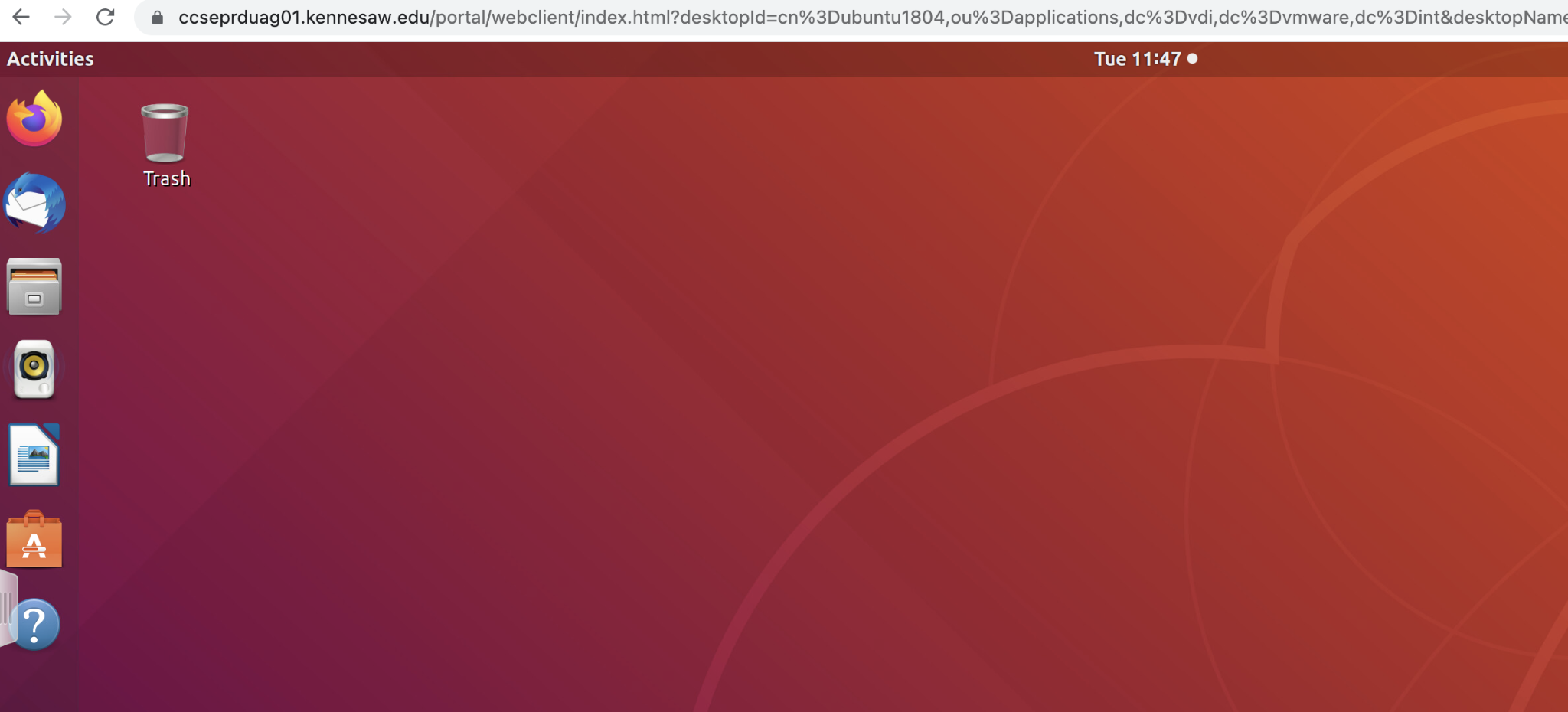
**Instructions**

We are going to use CSE Lab Ubuntu Virtual Machine for this lab.

* To access, please go to: <https://cseview.kennesaw.edu/>
  + You can download the VMware Horizon Client or use the HTML version
* Launch the VMware
* Select Ubuntu Virtual Machine and Launch it.



* You will see Ubuntu. Username: Administrator / Password: linuxadmin
* The interface will look like this

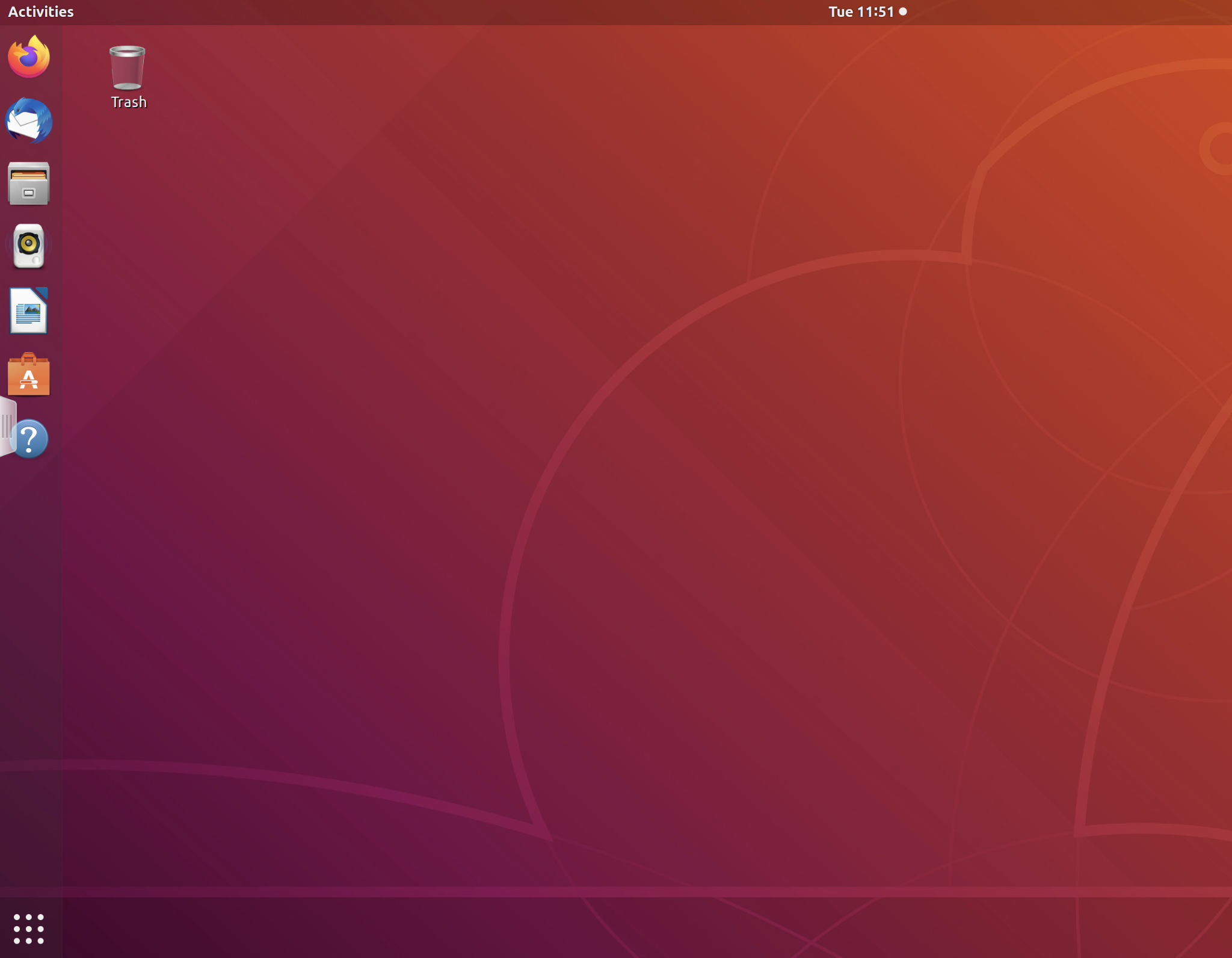


To conduct this lab, we need to install two programs

* Docker ( is an open source containerization technology for building and containerizing applications). To install Docker Engine, follow these steps;
  + Open a browser (Firefox, on the left top menu)
  + Type the following address <https://tinyurl.com/4333-installdocker> (This will download an script to install Docker Engine)
  + Keep the file there
* Kathara (which is a lightweight network emulation system based on Docker containers). To install Kathara follow these steps:
  + Open a browser (Firefox, on the left top menu)
  + Type the following address <https://drive.google.com/file/d/1_WSR_cV3g2WVrx8540a92UO5lfIEmzPi/view?usp=sharing> (This will download an script to install Docker Engine)
  + Keep the file there
* Download the lab files for this specific lab here: <https://github.com/KatharaFramework/Kathara-Labs/raw/master/Basic%20Topics/ARP/kathara-lab_arp.zip>
  + Decompress the file.

After you download all the require files, you must

* Open a terminal by clicking in the bottom left of the screen and searching “terminal” word

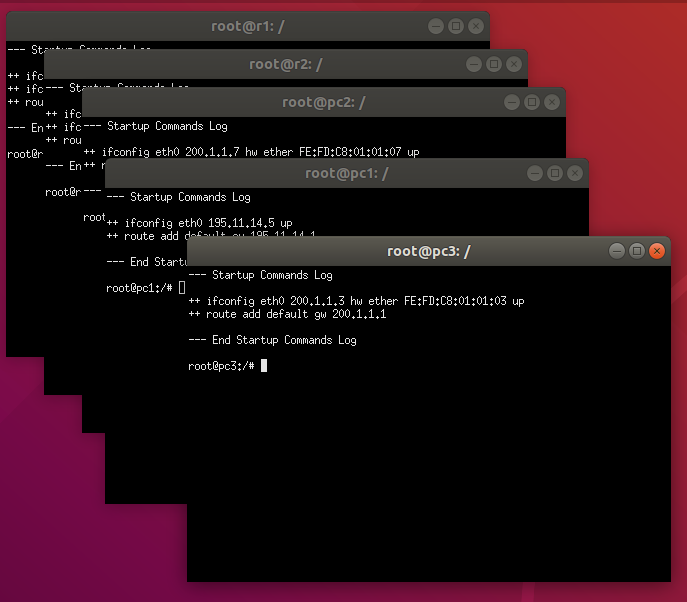




* On the terminal, using cd, go to the folder where the previous files (installsocket and installkathara) were downloaded (typically Download folder)
* Once there, type the following command
  + sudo chmod +x installsocket
  + ./installsocket
    - Note that this can take some time. Please check the screen for any questions. Always reply Yes for installation. If the password is requested, please use linuxadmin
  + sudo chmod +x installkathara
  + ./installkathara
    - Note that this can take some time. Please check the screen for any questions. Always reply Yes for installation. If the password is requested, please use linuxadmin

NOTE: Docker and Kathara installation will be permanent in your VM. You can log out and log in later and you do not need to start over. You must continue again in the following step.

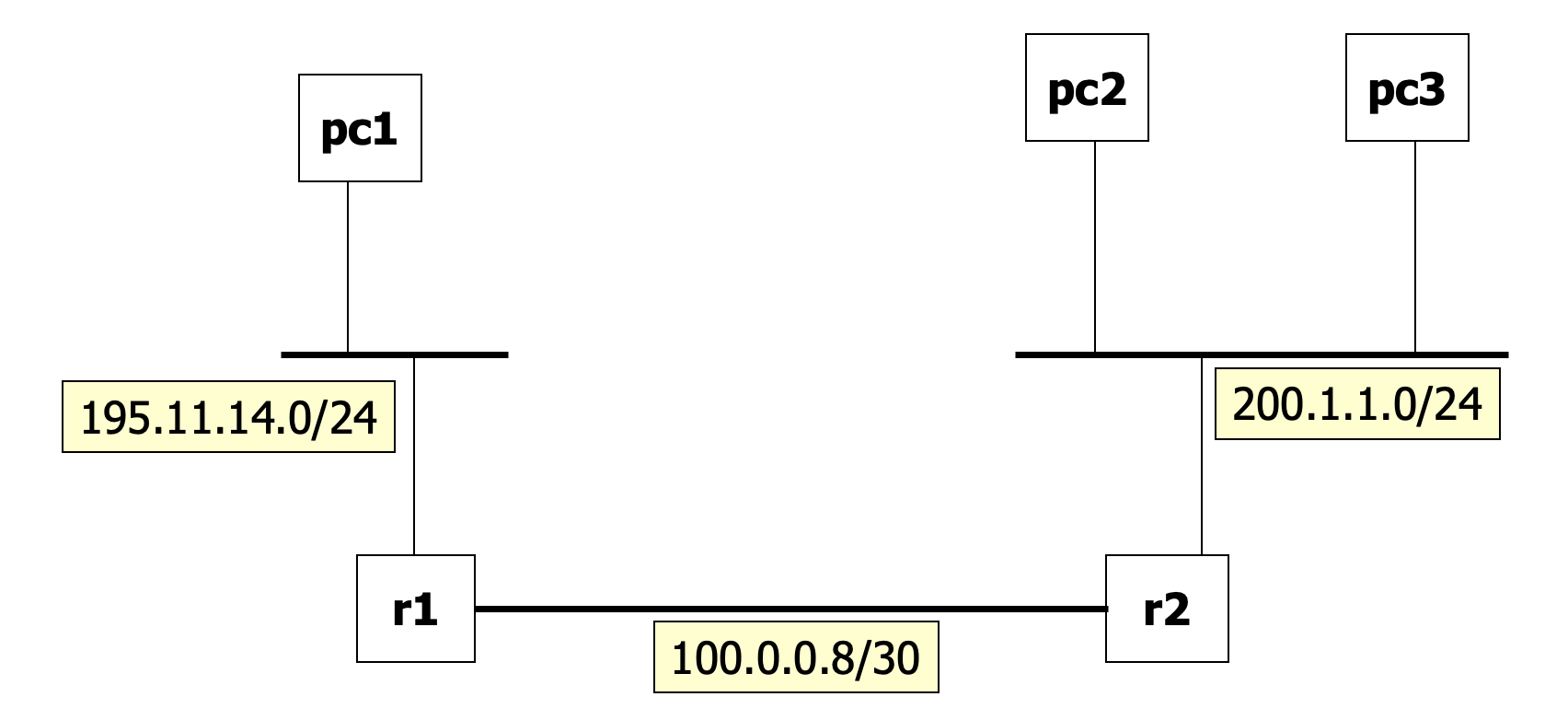
* Once Docker and Kathara are installed, please use cd to get into the folder where you decompress kathara-lab\_arp. Then,m type the following command
  + kathara lstart
    - If this doesn’t work, you can try using the sudo version: sudo kathara lstart
    - The process is long, and takes some minutes. Then, you will see a terminal for each machine (pc and router) in the lab



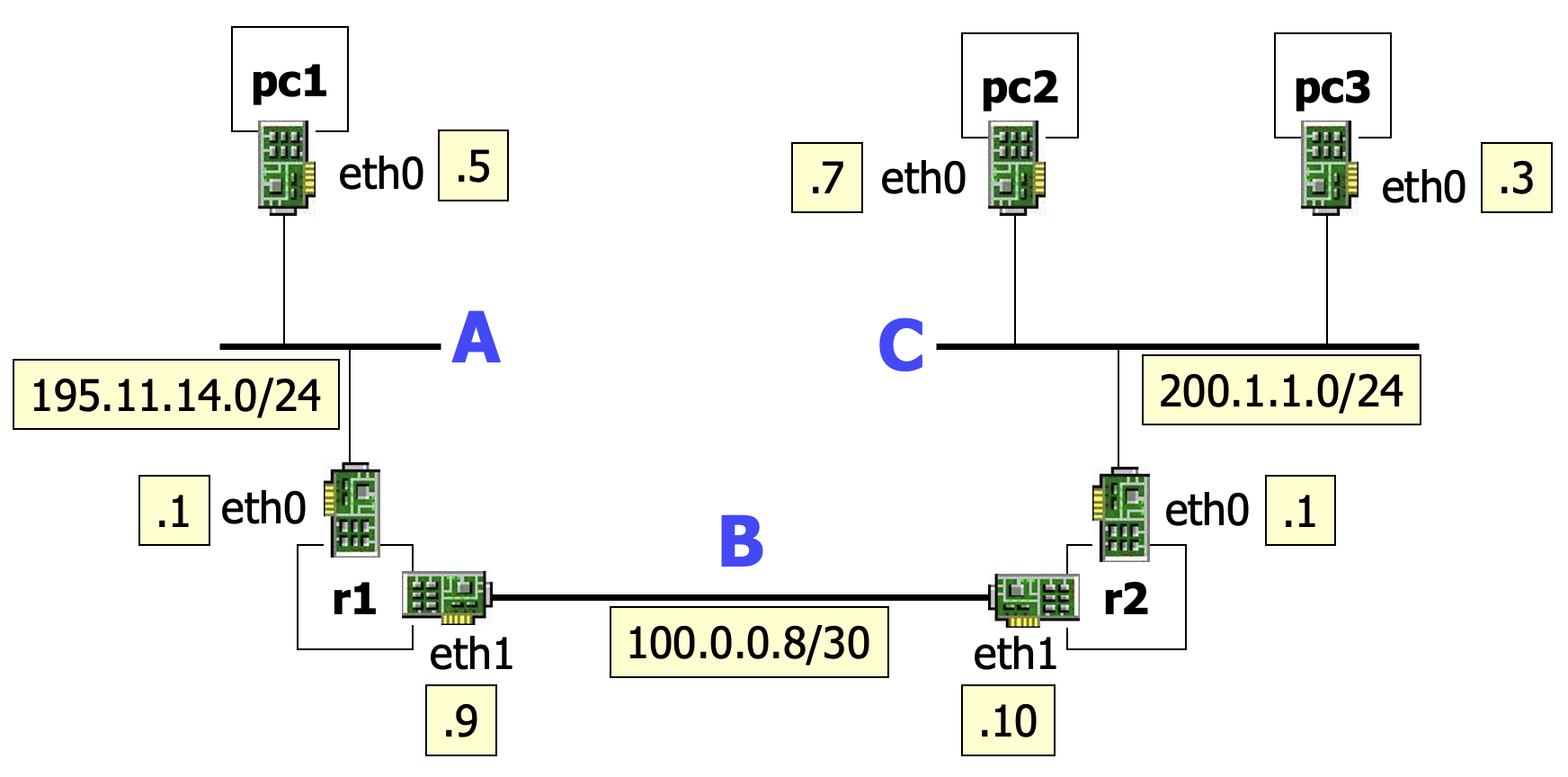
**PLEASE ADD A SCREENSHOT OF THIS STEP IN YOUR LAB (10 points)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**In this lab, we have the following network topology**

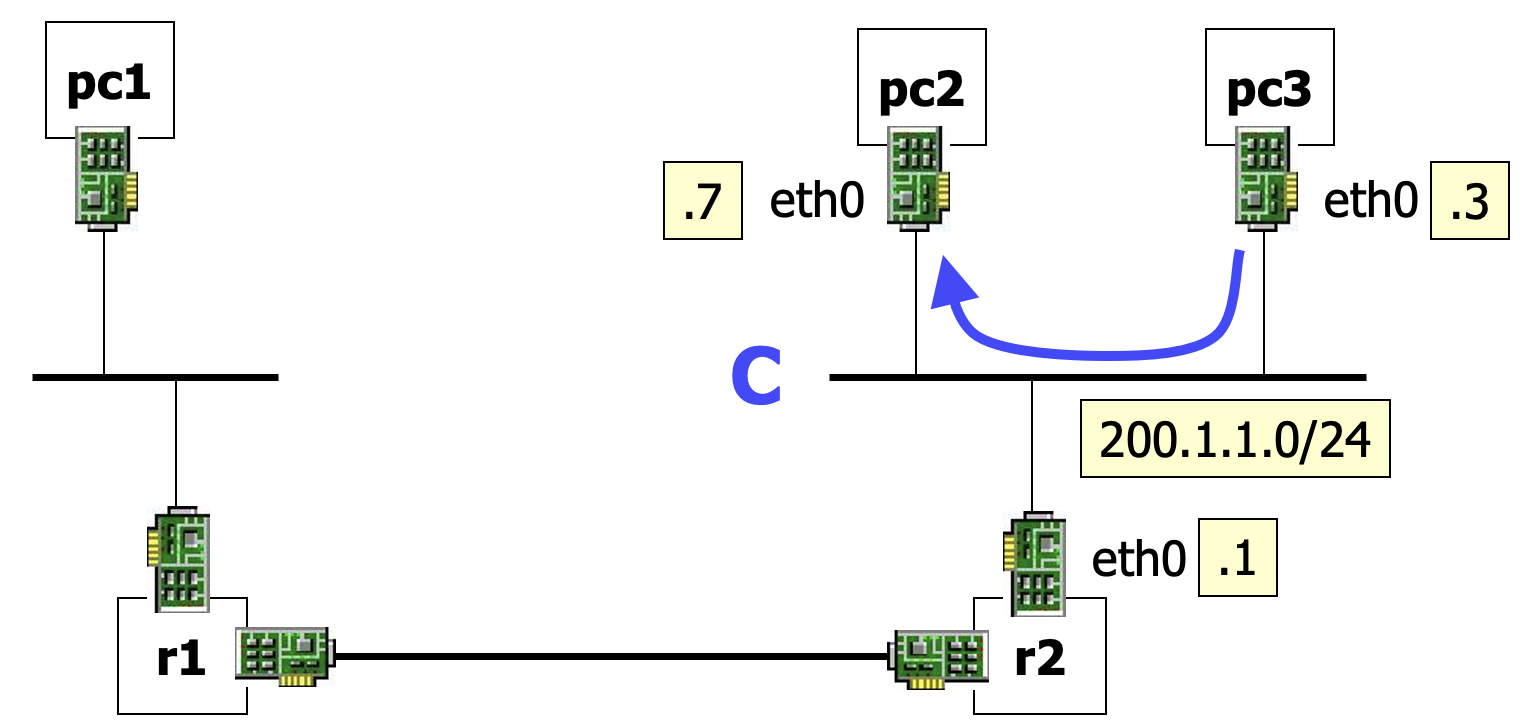
****

**These are the configuration details**

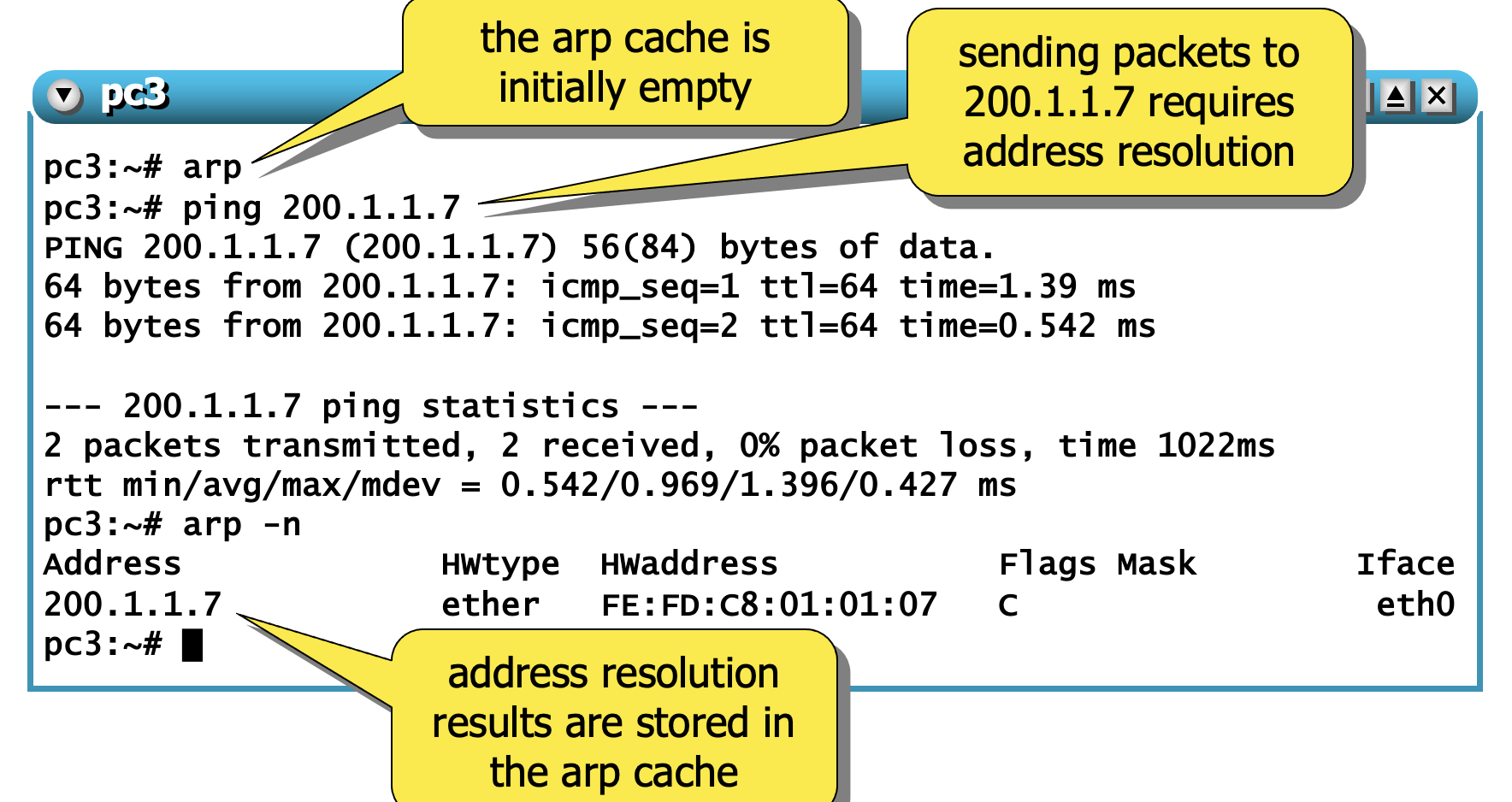
****

**Step 1. Inspecting ARP Cache**

1. **Traffic within the same network does not traverse routers**

****

1. **Type the following command in PC3 terminal** 
   1. **arp**
   2. **ping 200.1.1.7**
   3. **Stop the ping after some time using CTRL C**
   4. **arp -n**

****

1. **Please add a screenshot of Step 2 (10 points)**
2. **Communications are usually bidirectional. The receiver of the arp request learns the mac address of the other party, to avoid a new arp in opposite direction (standard behavior, see rfc 826)**

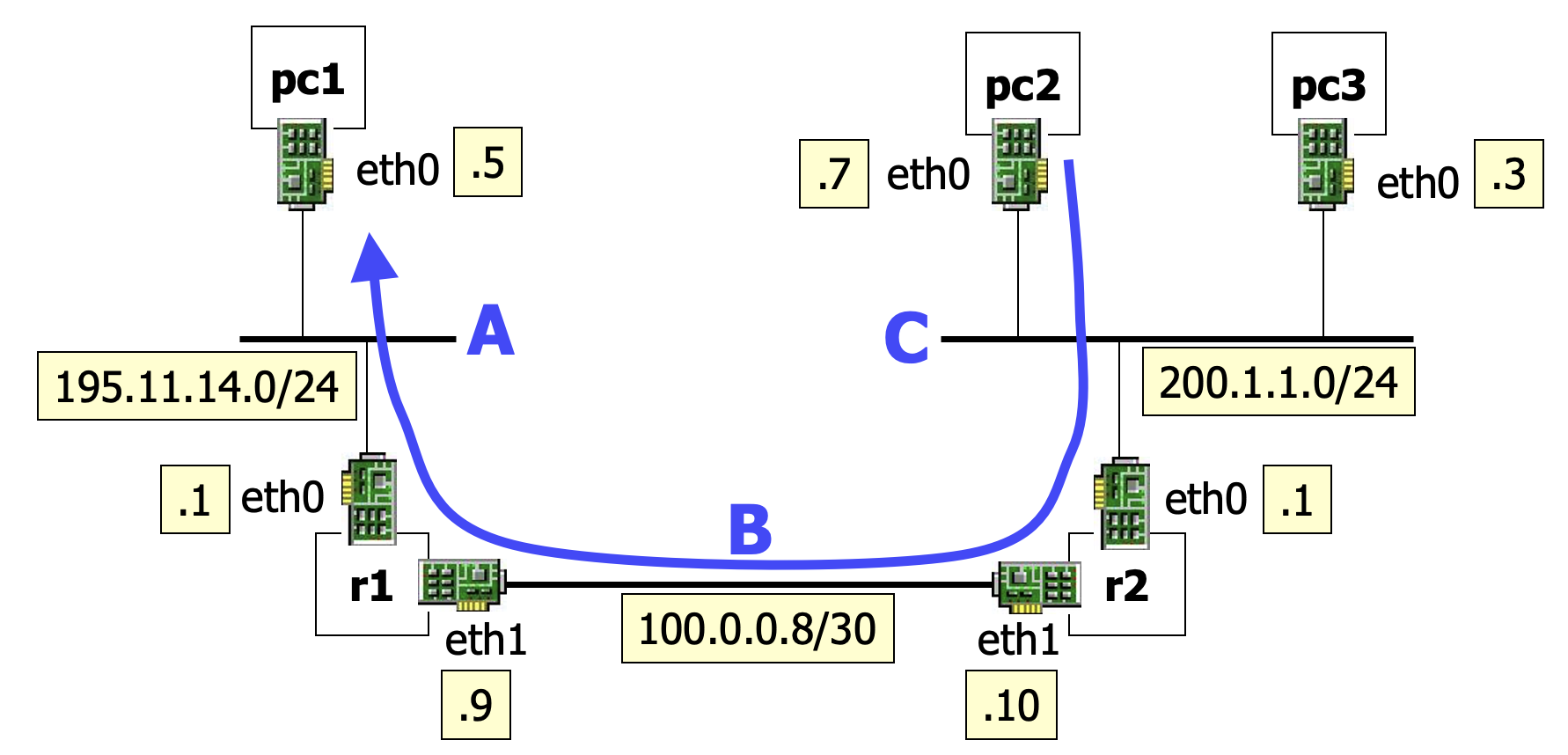
**Go to PC2 terminal and type**

1. **arp -n**

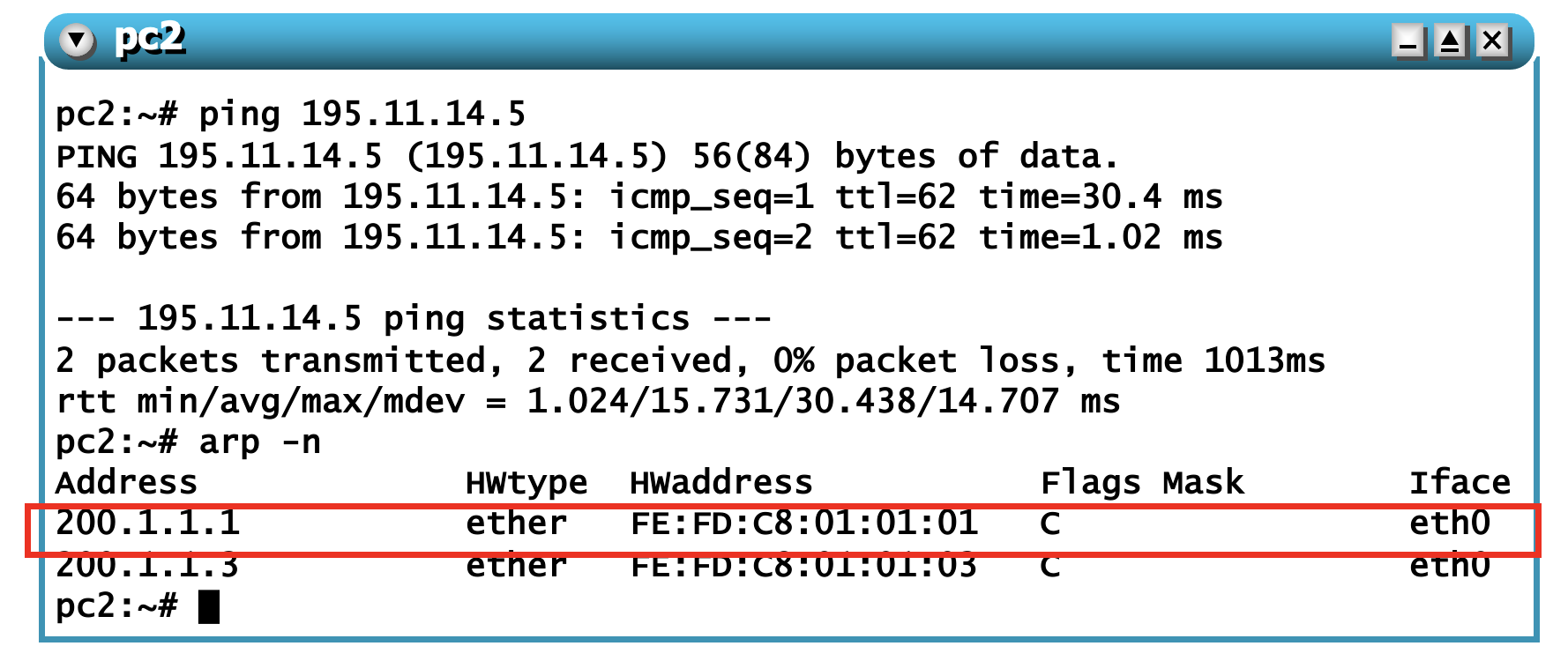
****

1. **Please add a screenshot of Step 4 (10 points)**

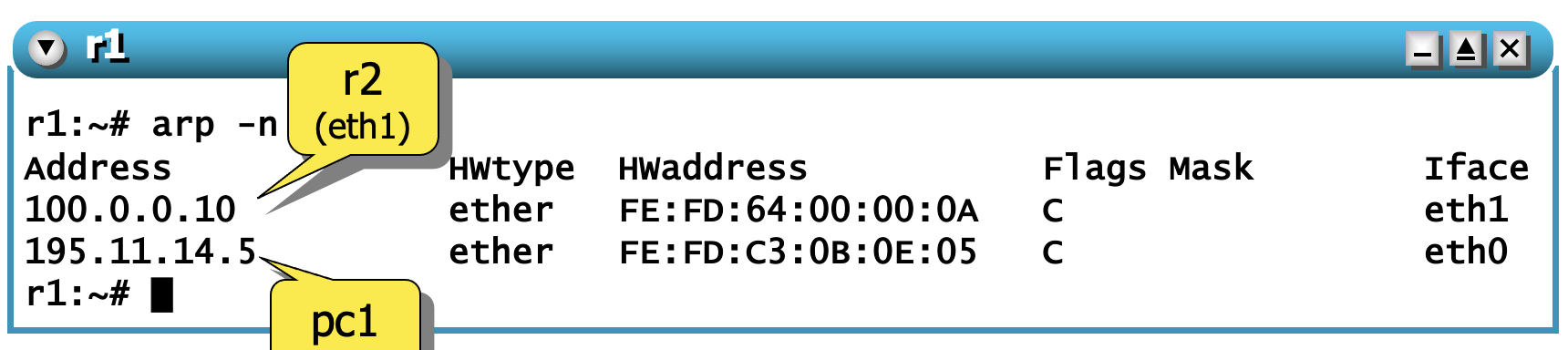
**Step 2. Inspecting ARP Cache (non-local traffic)**

****

1. **When ip traffic is addressed outside the local network, the sender needs the mac address of the router. ARP request can replies only within the local network**
2. **In the PC2 terminal type the following commands**
   1. **Ping 195.11.4.5**
   2. **Stop the ping after some time using CTRL C**
   3. **arp -n**

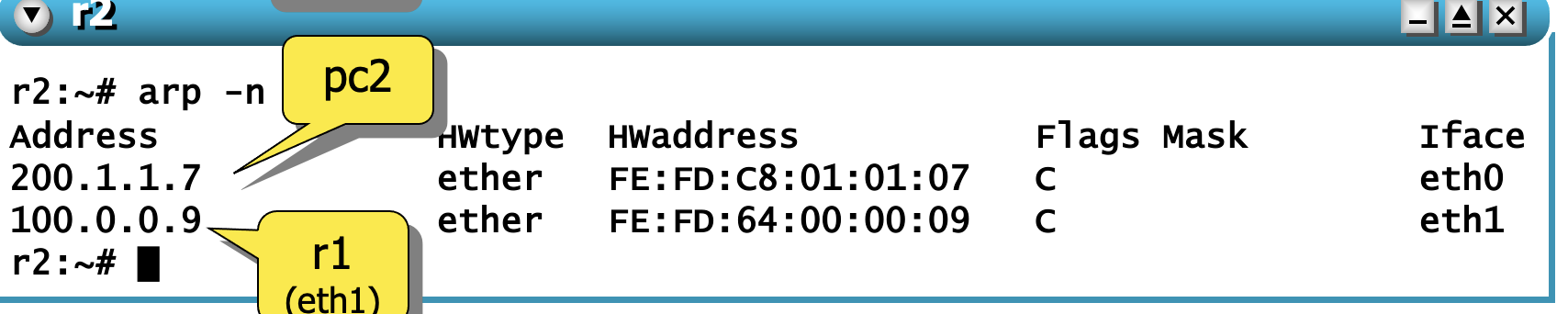
****

1. **Who is 200.1.1.1 in the previous result? (5 points)**
2. **Go to terminal R1 and type**
   1. **Arp -n**

****

* 1. **Please add a screenshot of Step 4a (5 points)**

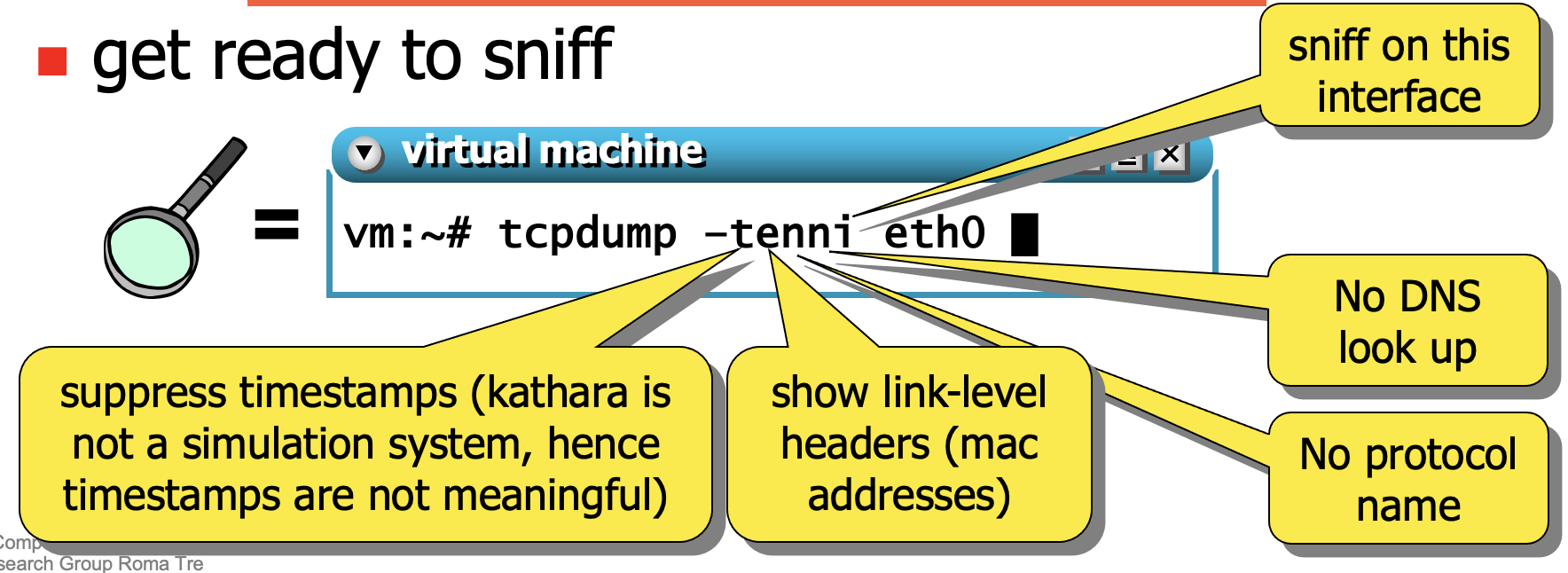
1. **Go to terminal R2**
   1. **Arp -n**

****

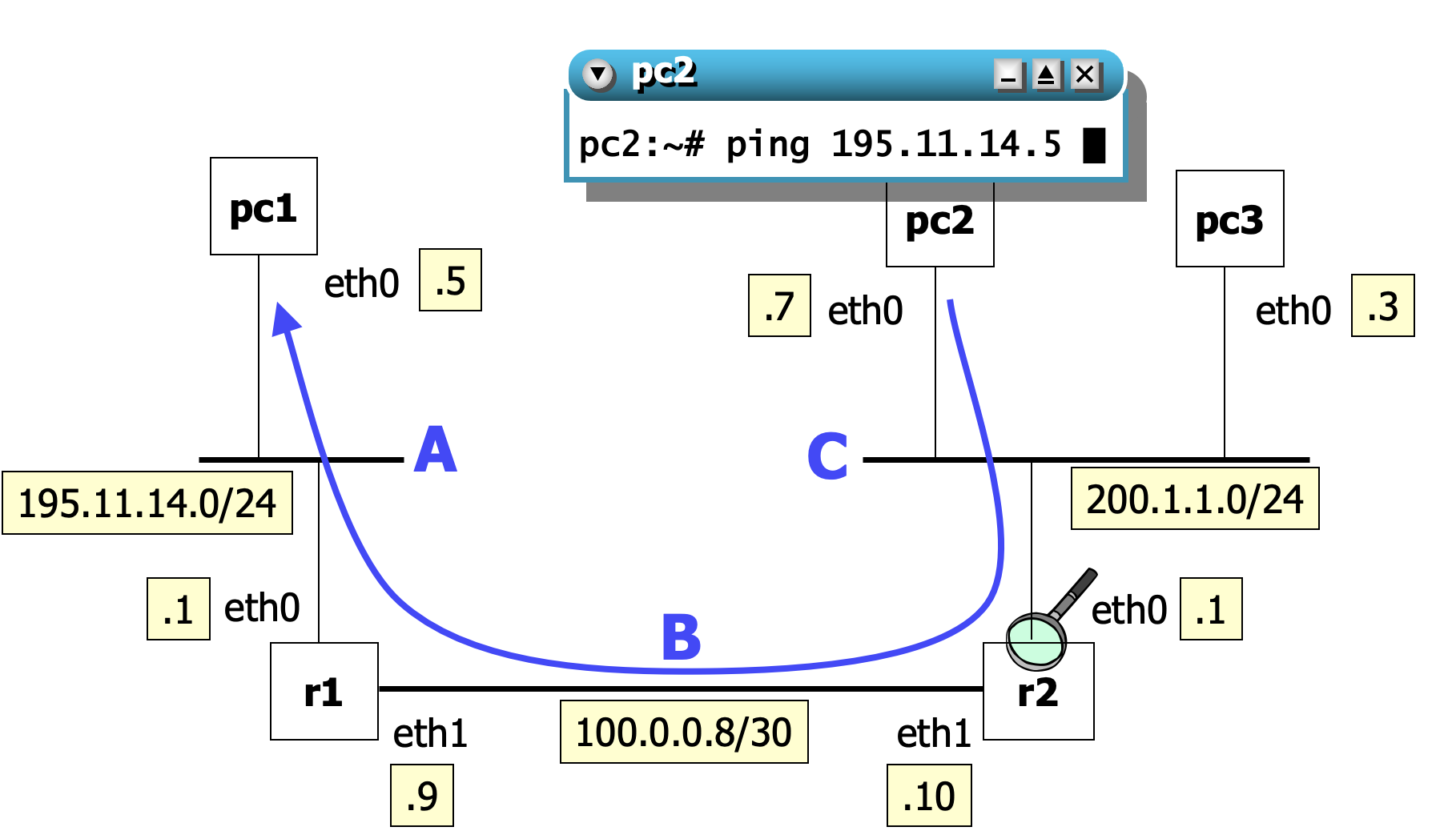
* 1. **Please add a screenshot of Step 5a (5 points)**

**Step 3. Sniffing ARP traffic**

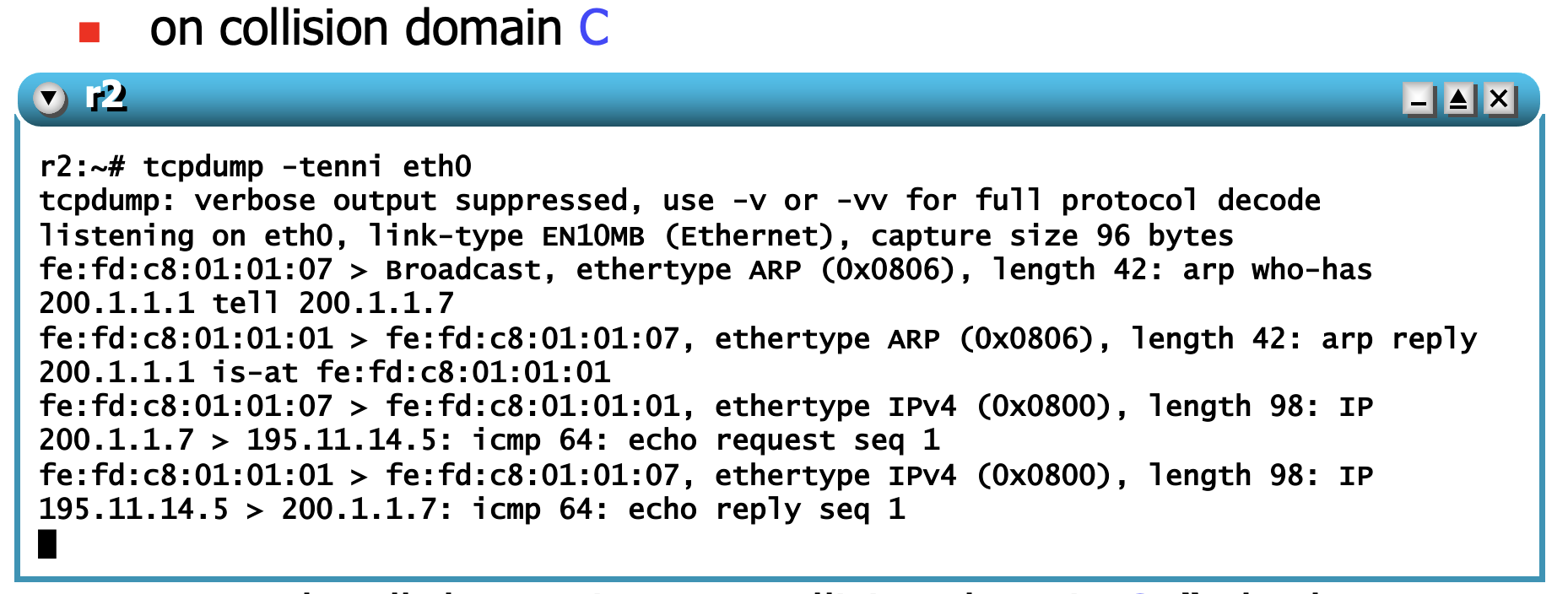
1. **Close all PCs and Rs terminals**
2. **Restart the lab to clear ARP cache**
   1. **kathara lclean (or sudo kathara lclean)**
   2. **kathara lstart (or sudo kathara lstart)**
3. **Get ready to sniff**

****

1. **In PC2, do a ping to 195.11.14.5**

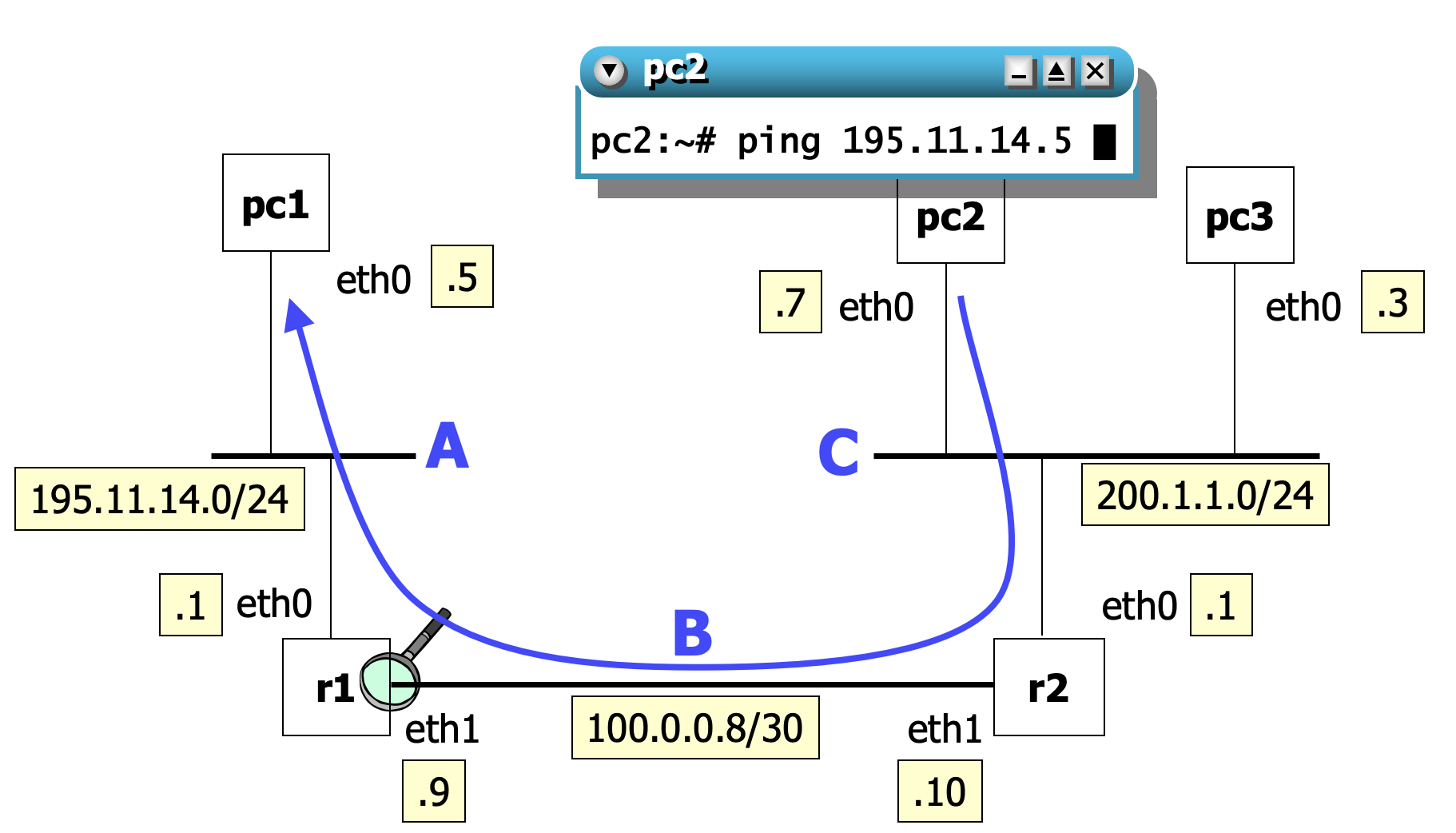
****

1. **Let’s see in detail what is happening in r2. Open r2**

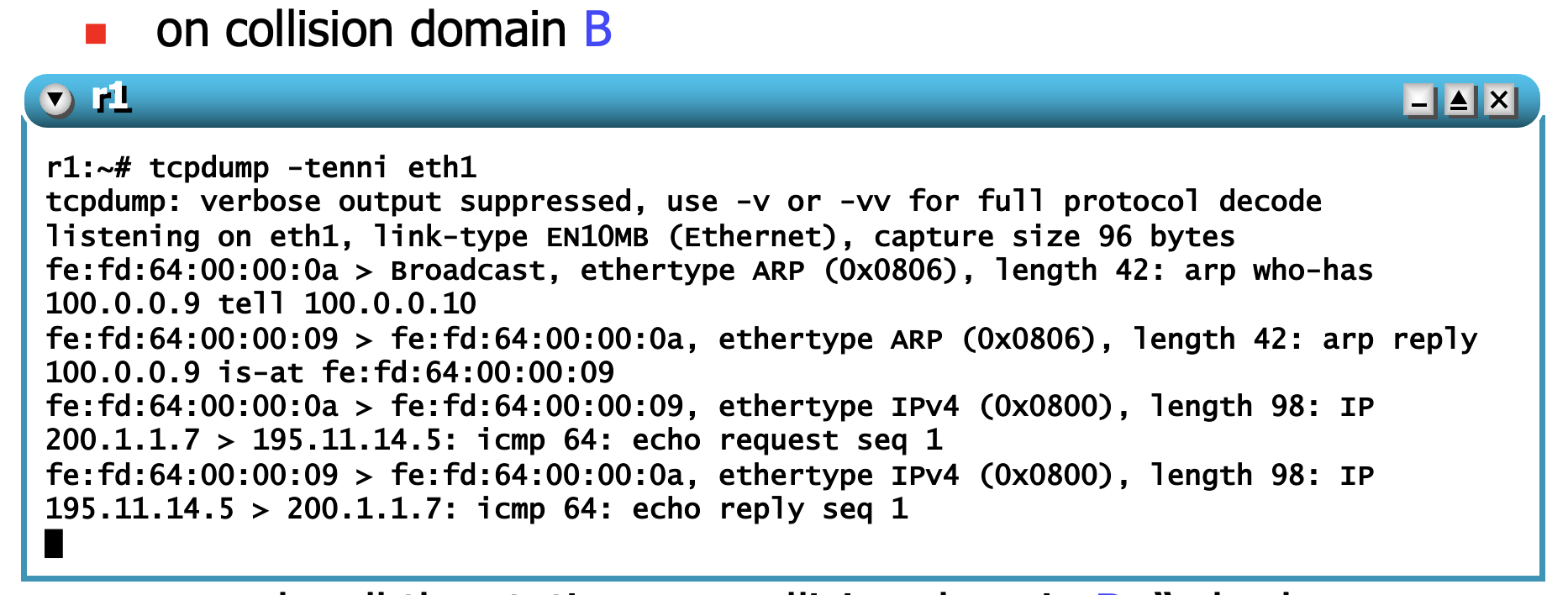
****

* 1. **This is what is happening here**
     1. **PC2 asks all the stations on collision domain C: “who has 200.1.1.1?” (200.1.1.1 is pc2’s default gateway)**
     2. **r2 replies → both pc2 and r2 update their arp cache**
     3. **pc2 sends to r2 the ip packet (icmp echo request) for pc1**
     4. **r2 sends to pc2 the corresponding echo reply (generated by pc1)**
     5. **Please add a screenshot of Step 5 (5 points)**

1. **Let’s see in detail what is happening in r1.**

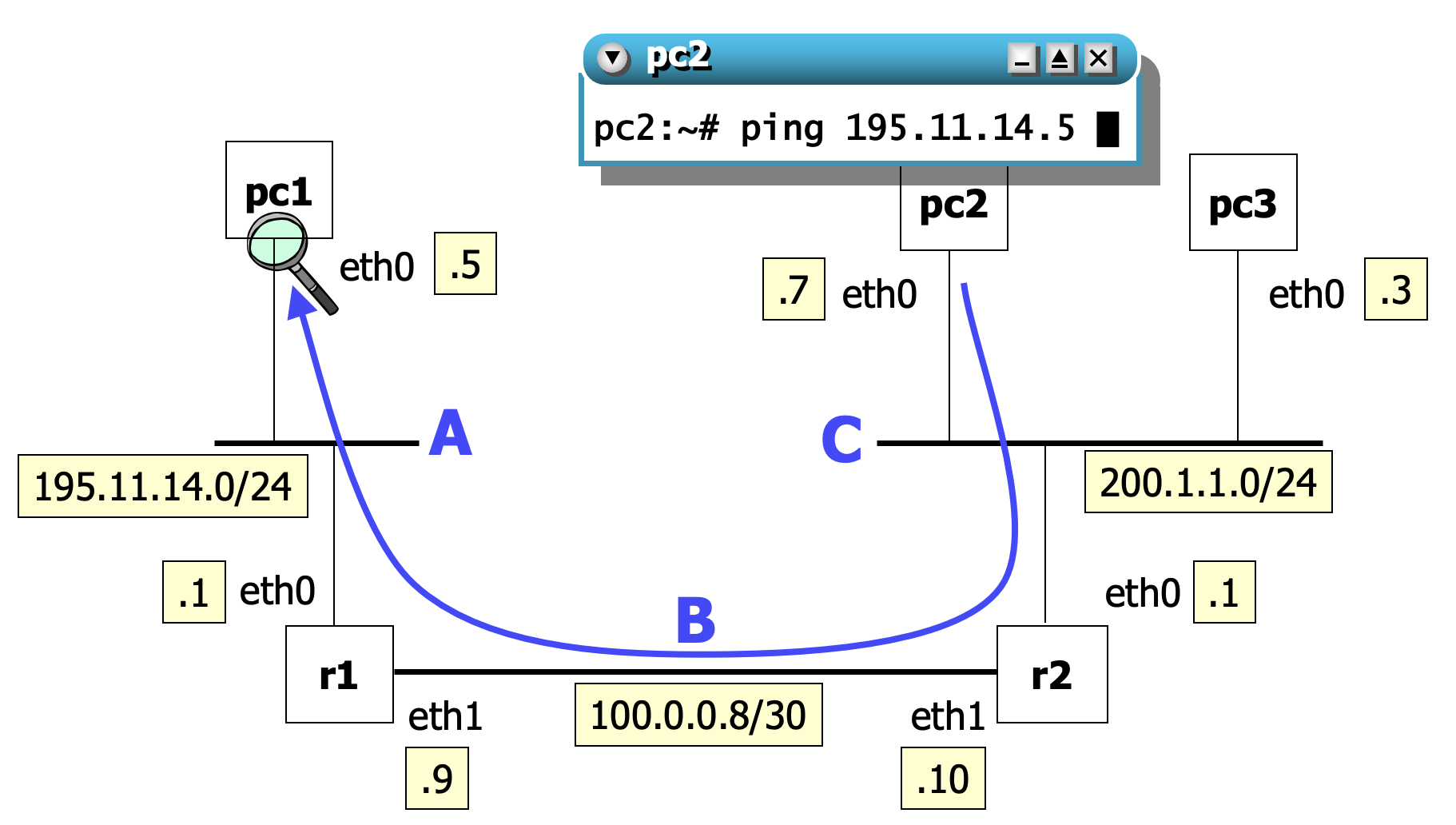
****

* 1. **Open r1 and type → tcpdump -tenni eth0**

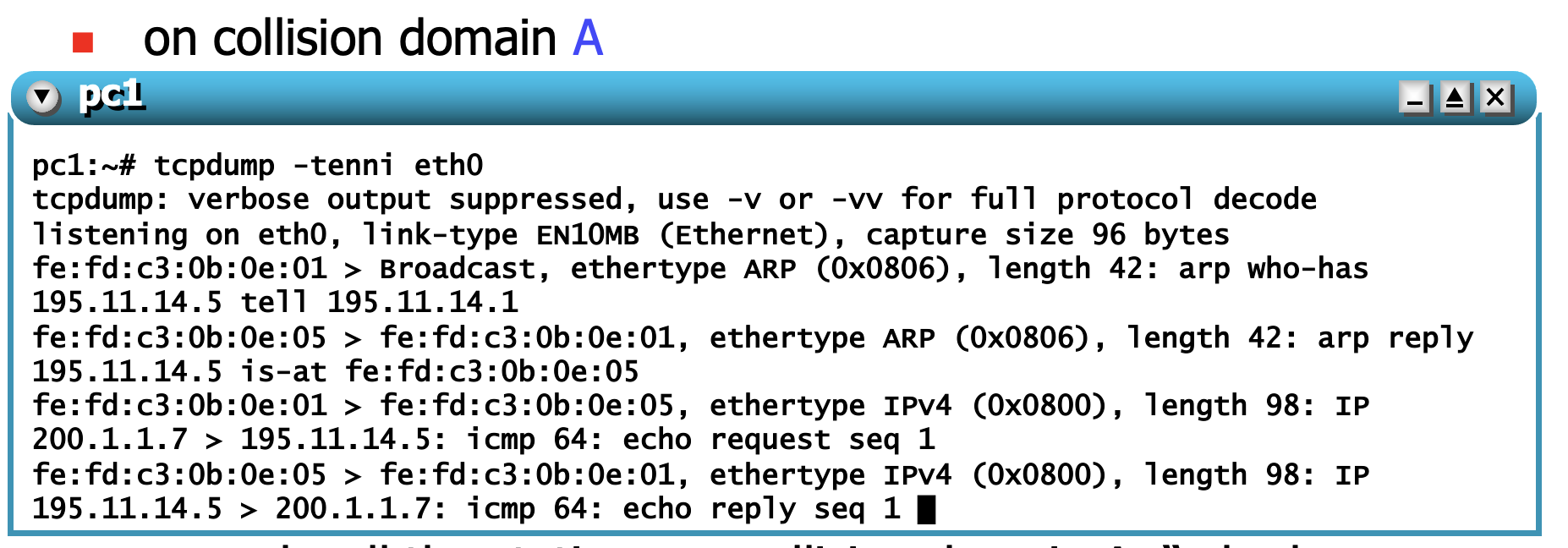
****

* + 1. **r2 asks all the stations on collision domain B: “who has 100.0.0.9?” (100.0.0.9 is the next hop obtained from the routing table)**
    2. **r1 replies → both r1 and r2 update their arp cache**
    3. **r2 sends to r1 the echo request generated by pc2 for pc1**
    4. **r1 sends to r2 the echo reply generated by pc1 for pc2**
    5. **Please add a screenshot of Step 6 (5 points)**

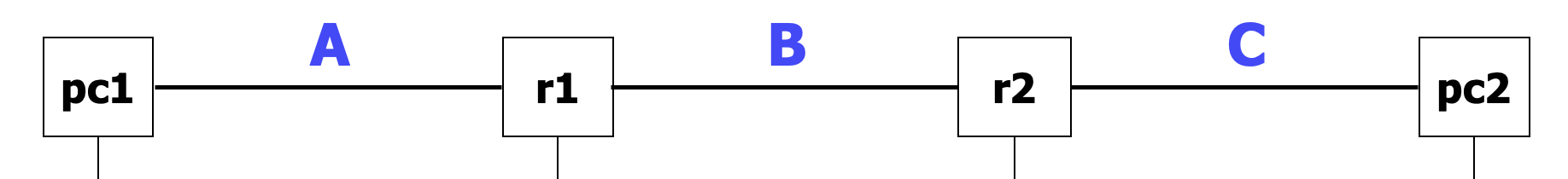
1. **Let’s see in detail what is happening in PC1**

****

* 1. **Open PC1 and type → tcpdump -tenni eth0**

****

* + 1. **r1 asks all the stations on collision domain A: “who has 195.11.14.5?” (195.11.14.5 is the destination address of the icmp request obtained from the ip header)**
    2. **pc1 replies --> both pc1 and r1 update their arp cache**
    3. **r1 sends the ip packet (echo request) to pc1**
    4. **pc1 generates the corresponding echo reply for pc2 and sends it to r1**
    5. **Please add a screenshot of Step 7 (5 points)**

1. **Please create a graph with all the message exchanged by the PCs and Rs (20 points)**
2. **check the different error messages (add screenshot) obtained by trying to ping an unreachable destination in the case of**
   1. **nlocal destination (10 points)**
   2. **nnon local destination (10 points)**