CSCI 5180 – Network Management

and Automation

Lab 2

DHCPv4, DHCPv6 - Auto-configuration, Prefix Delegation & Scapy

University of Colorado Boulder

Department of Computer Science

Network Engineering

Professor Levi Perigo, Ph.D.

# Summary Introduction to DHCPv4 (Scripting), DHCP Auto-configuration and Prefix-Delegation

This lab focuses on DHCP and provides an insight into IP address configuration in a routed network. We will be using GNS3 to perform all the objectives in this lab. The first part of the lab focuses on configuring DHCPv4 through scripting, and the remaining objectives are intended to instill strong foundation knowledge on DHCPv6. DHCPv6 will be used for auto-configuration of IP addresses and delegating prefixes. In the final part of the lab, different packets are constructed through Scapy.

Objectives  
  
PART 1: DHCPv4 Scripting [45 Points]

1. Boot-up the Virtual Machine provided to you for this course. Start GNS3 and create the topology as shown below.

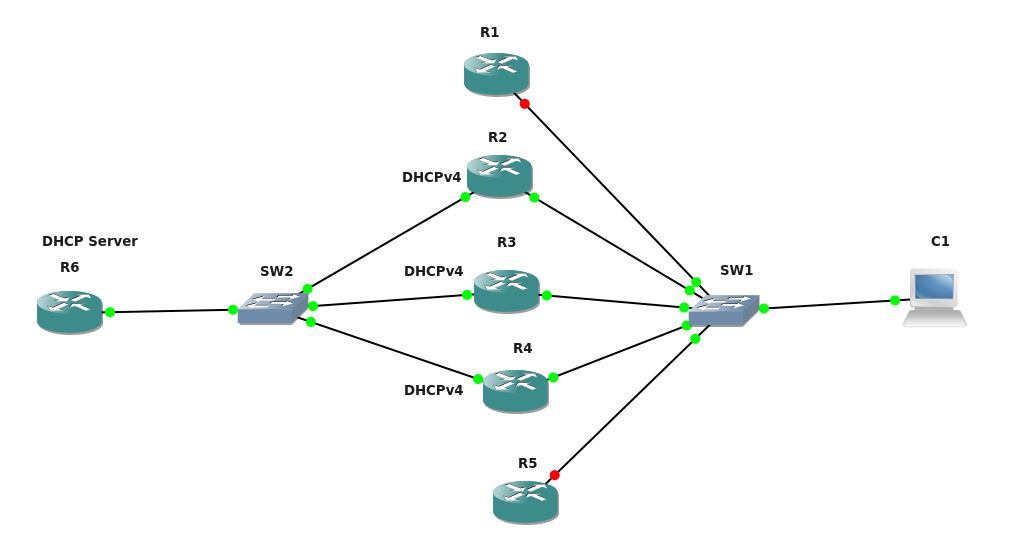
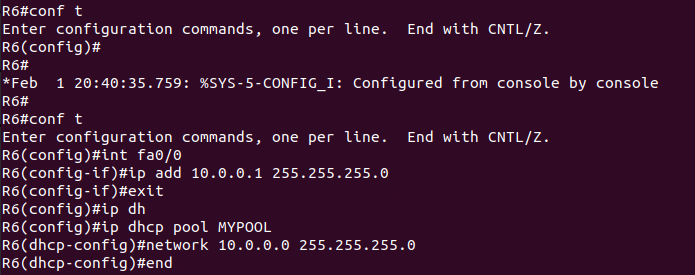
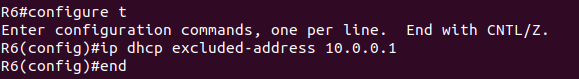
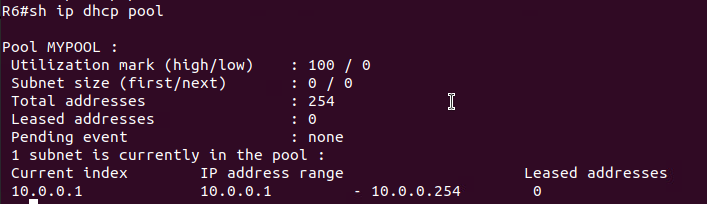


Fig. 1

1. Configure R6 to act as a DHCP Server serving IPv4 addresses. Paste relevant screenshot of the configuration. **[5 points]**

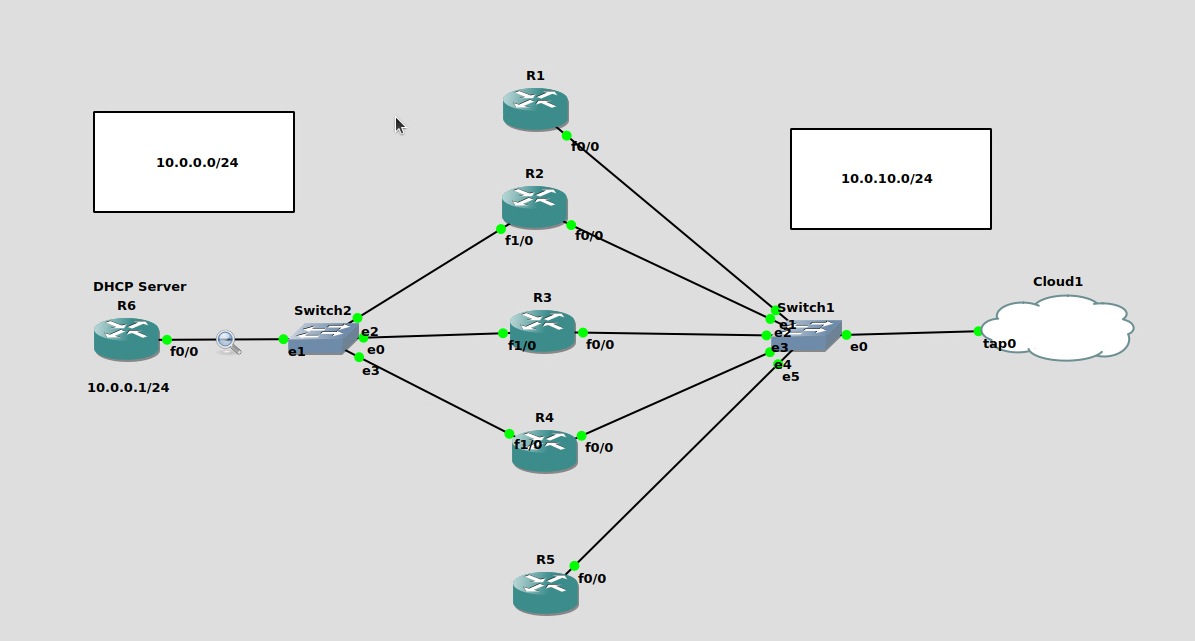




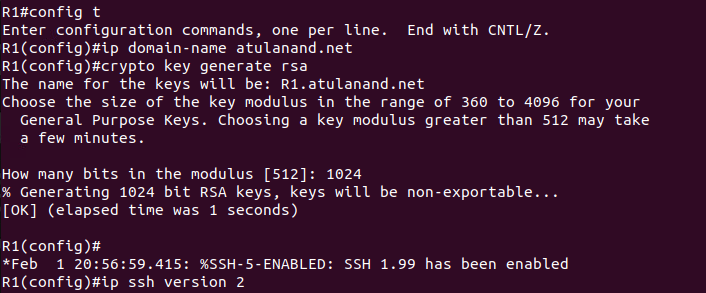


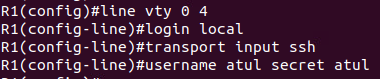
1. Write a Python script **DHCPv4Config.py** to configure DHCPv4 for the interfaces of R2, R3, R4 (facing SW2) to obtain IPv4 addresses from the DHCPv4 server (R6). Your code should facilitate simultaneous login to all three routers and deploy the configuration concurrently. Paste relevant screenshots of the output and submit the script along with the report. **[20 points]**

Note: - Please make sure that the DHCP pool subnet you define is different from your existing VM subnet. (198.51.100.0/24)

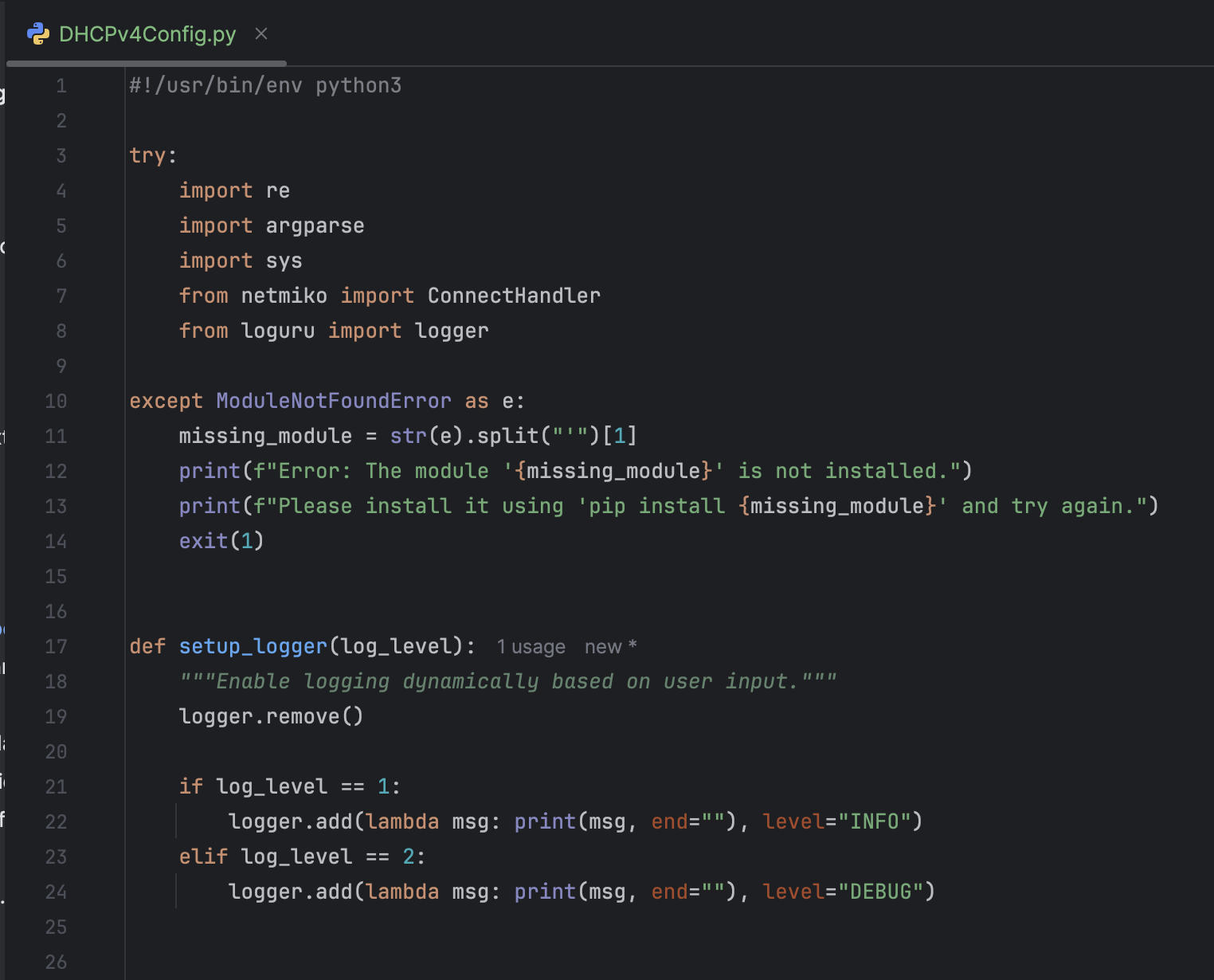


Configuring SSH on R1, R2, R3 and R4 –

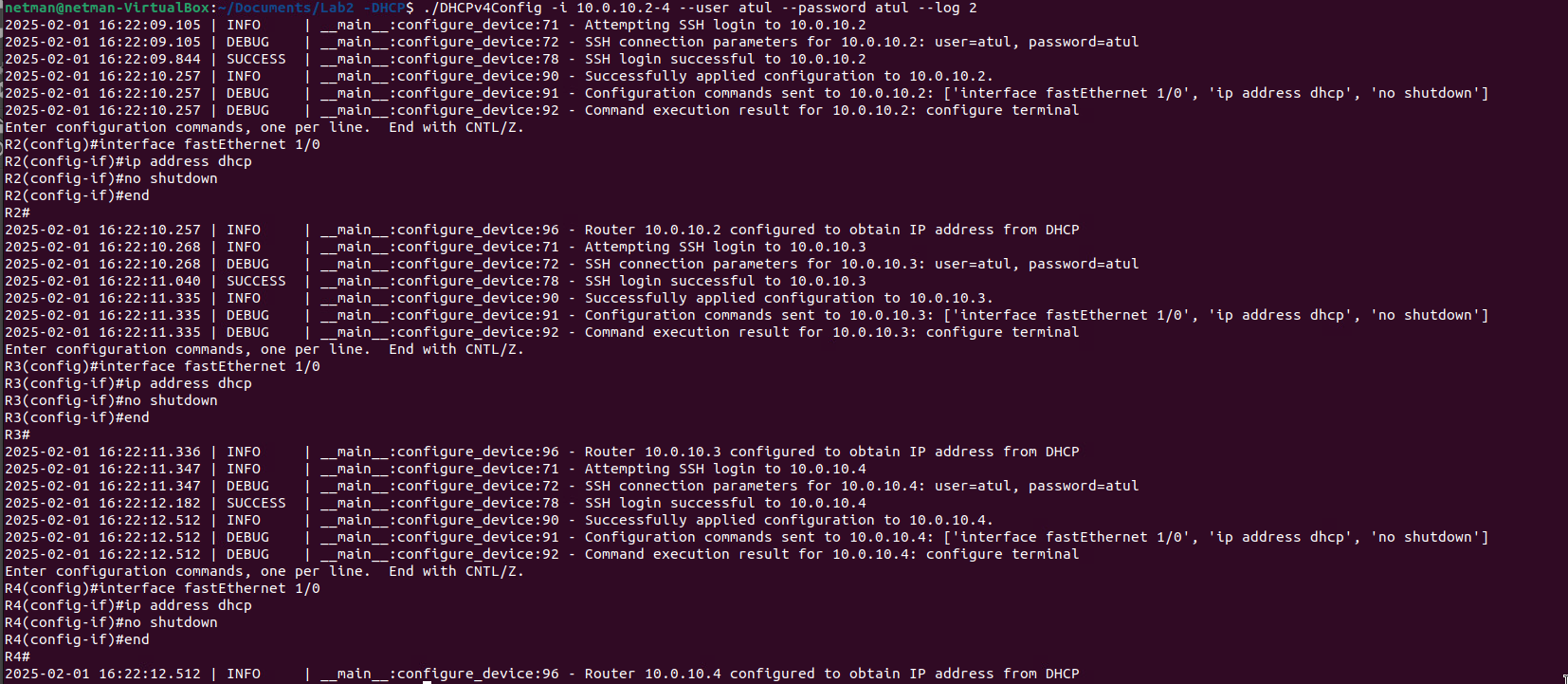




I have attached the DHCPv4Config.py script on the Canvas -

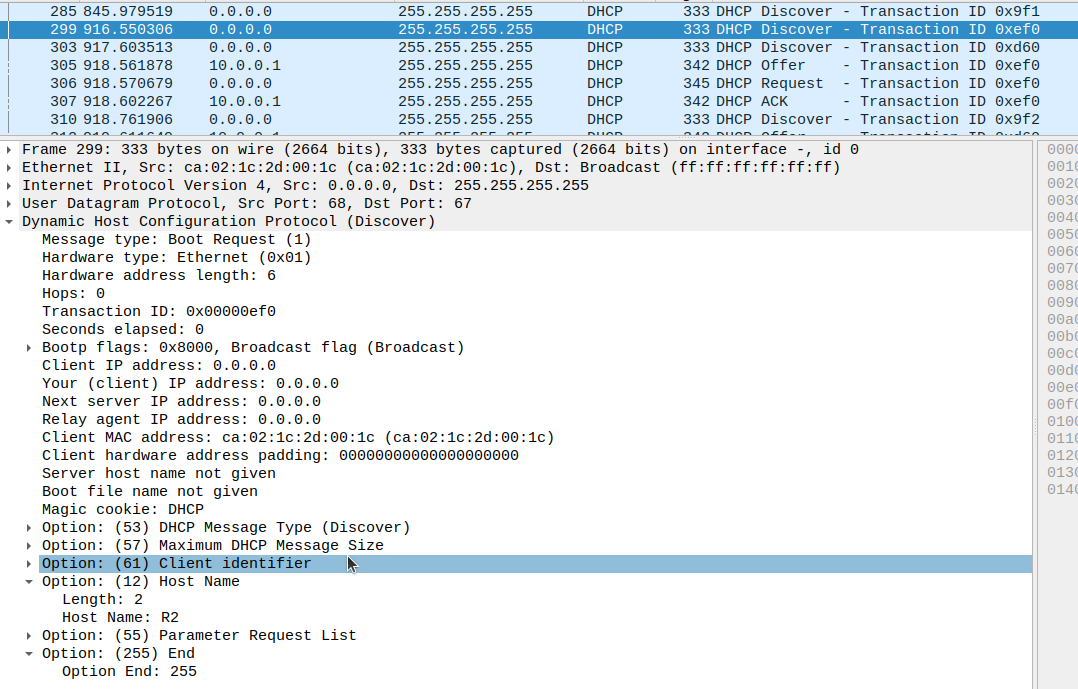


Output –



1. Monitor the interface connecting Switch SW2 and the DHCPv4 server (R6) and capture the DHCPv4 DORA messages using Wireshark:
   1. Use appropriate display filter to show the relevant DHCP messages between R2 and R6. Provide screenshots of the expanded packet view of the DORA messages. **[10 points]**

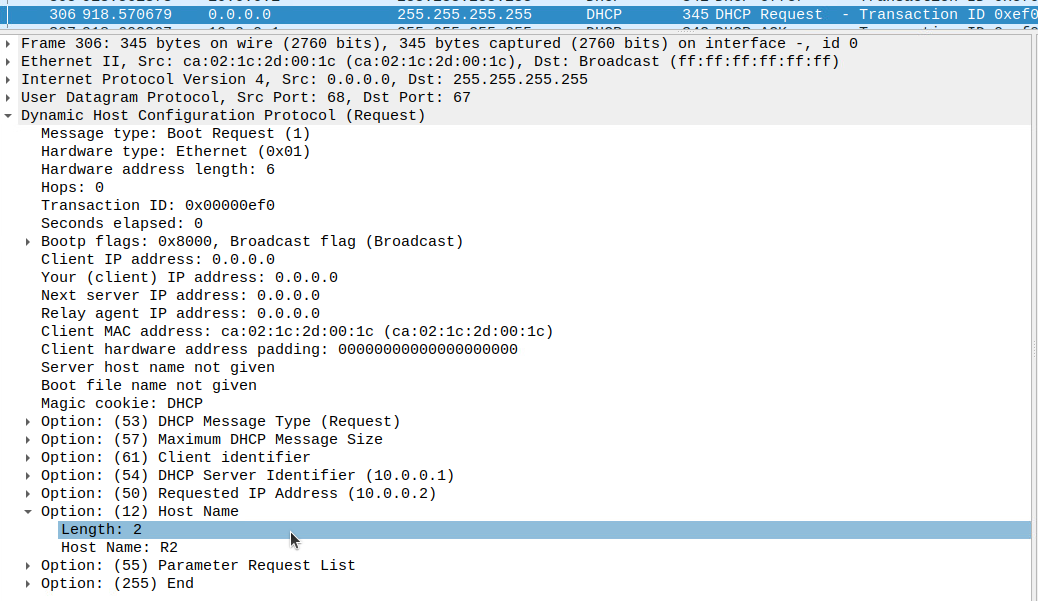
DHCP Discover (R2)



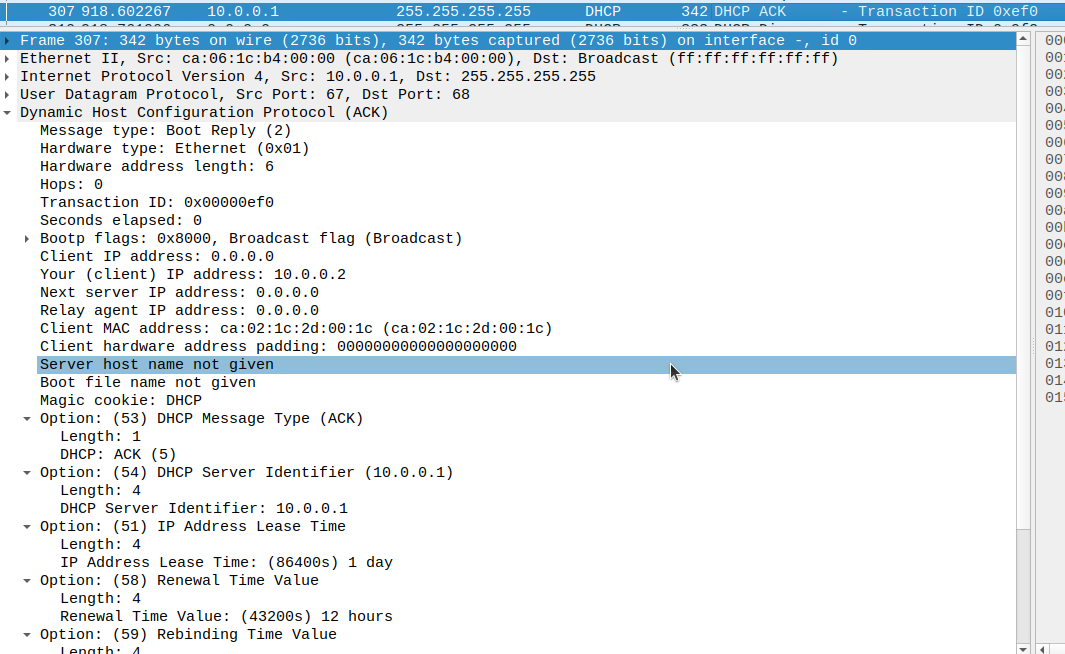
DHCP Offer –



DHCP Request –

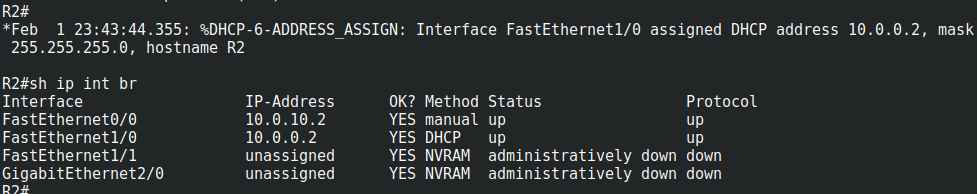


DHCP ACK –

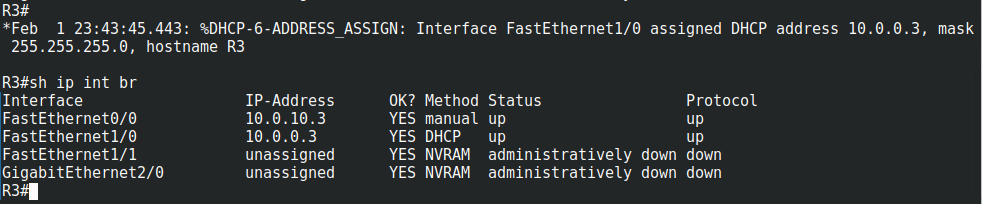


1. Provide a screenshot of the ‘show ip interface brief’ command on R2, R3, and R4 and ‘show ip dhcp binding’ on R6. **[5 points]**

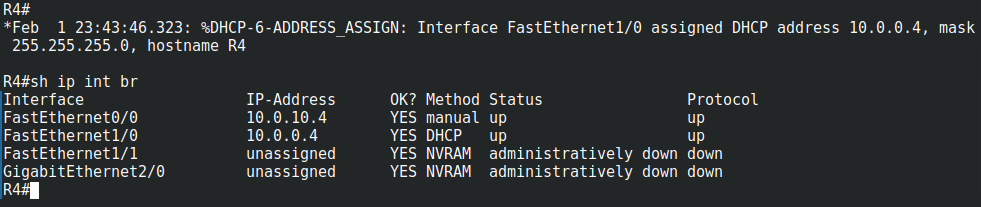
R2 –



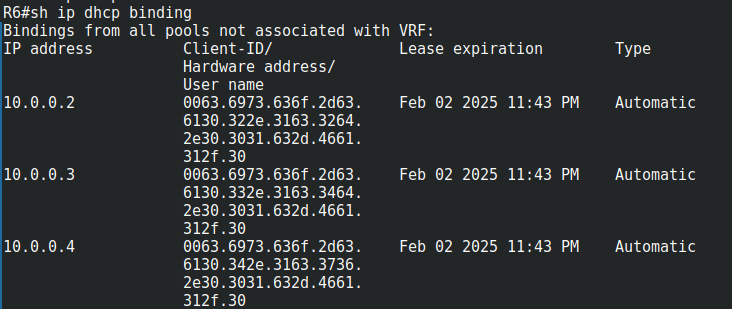
R3 –



R4 –

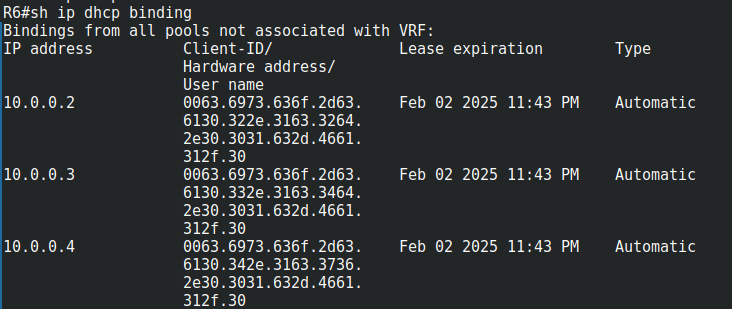


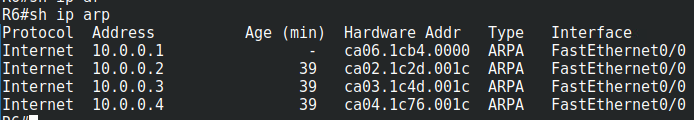
R6 –



* 1. Does the hardware address from the output of ‘show ip dhcp binding’ match the MAC addresses of the interfaces of R2, R3, and R4? Show how you can verify this. **[5 points]**

R6 (DHCP Binding)





R2 – Hardware Address-



R3 – Hardware Address-

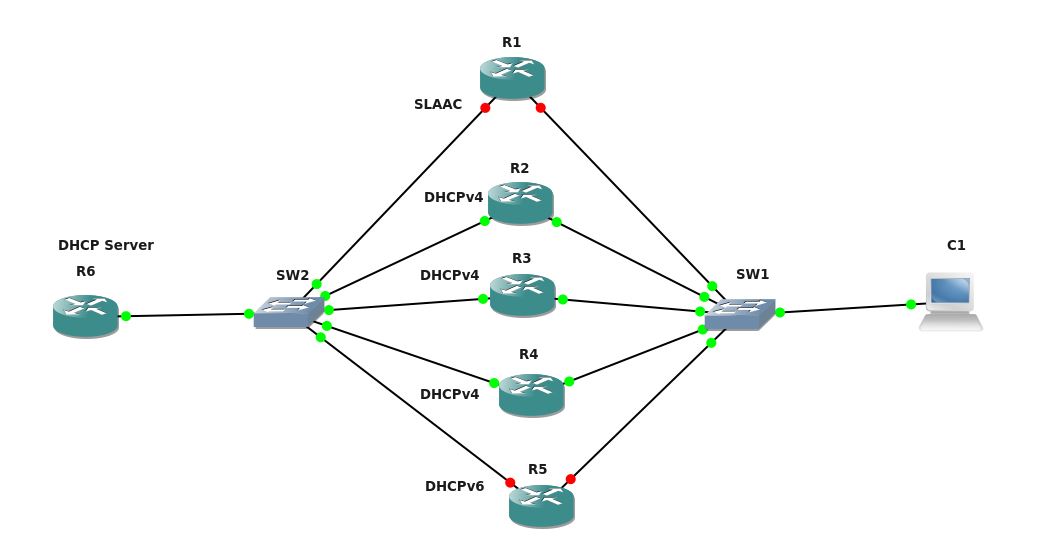


R4 – Hardware Address-



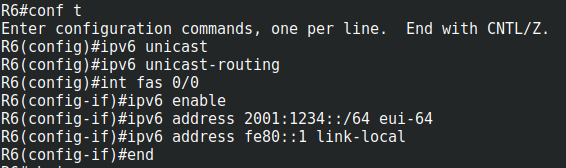
PART 2: DHCPv6 Auto Configuration [25 Points]  
  
IPv6 Auto-configuration Using SLAAC.

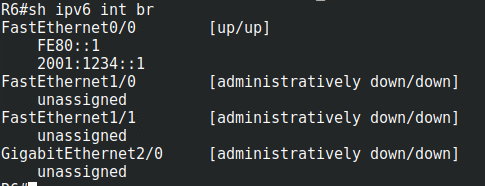
1. Configure the interface of R1 connected to the switch (SW2) to obtain an IPv6 address using SLAAC. **[10 points]**



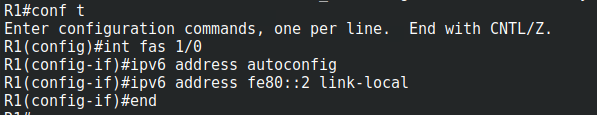
1. Provide a screenshot of the IPv6 configuration commands (on R6 and R1, as well as the Wireshark capture that shows R1 obtaining an IPv6 auto-configuration address. **[10 points]**

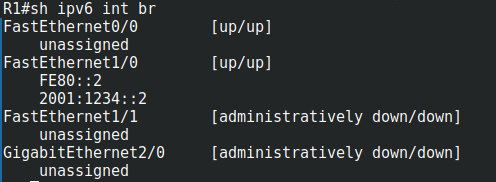
R6 –





R1 –

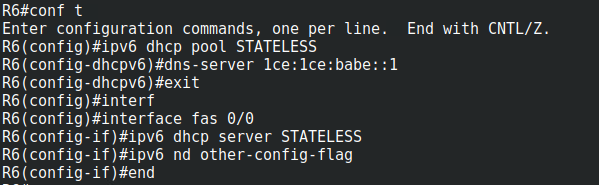




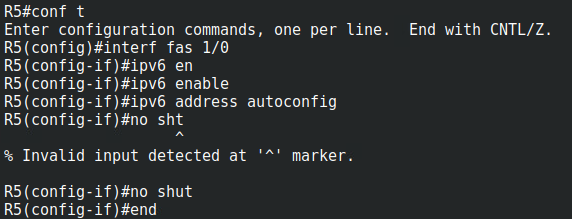


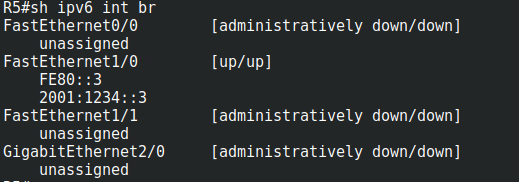
1. Configure R5 for DHCPv6 stateless. R5 should receive the DNS server IP address of 1ce:1ce:babe::1 from the DHCPv6 server (R6).

R6 –

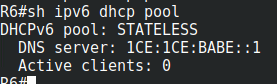


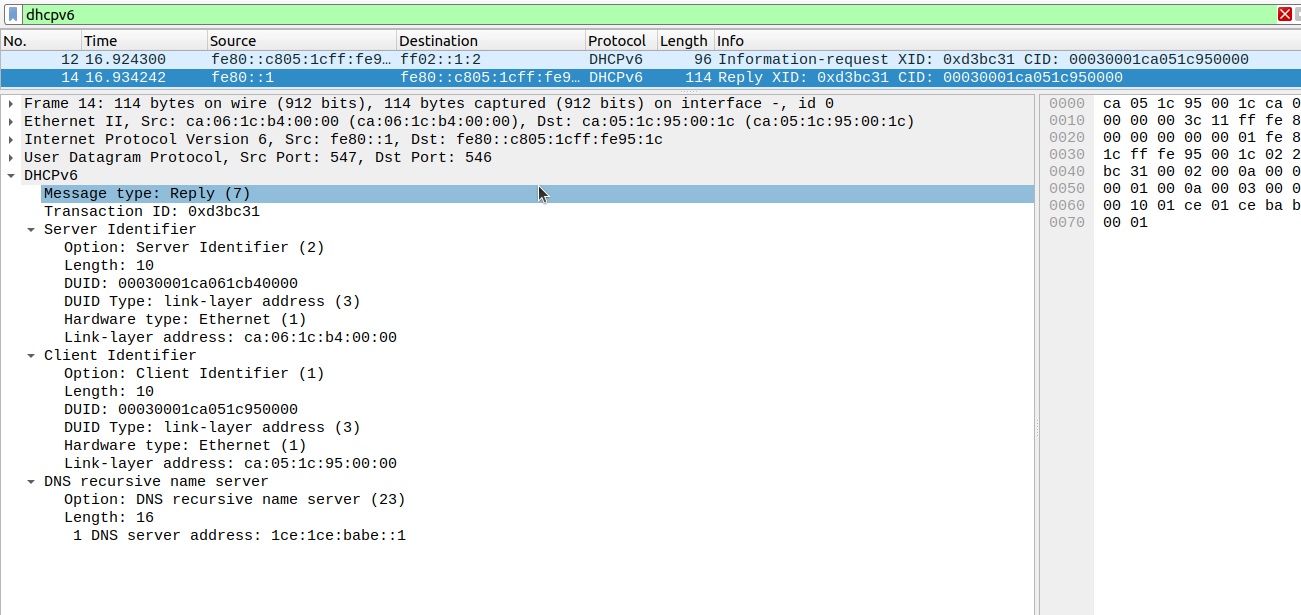
R5 –

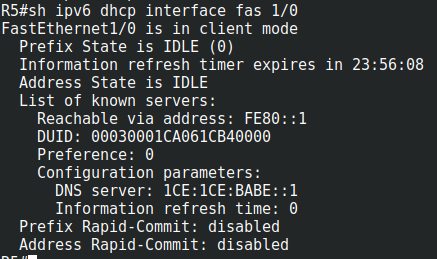




1. Provide a screenshot of the DHCPv6 stateless configuration on R6, and the Wireshark output from R5 indicating it received the DNS address via DHCPv6. [**5 points**]



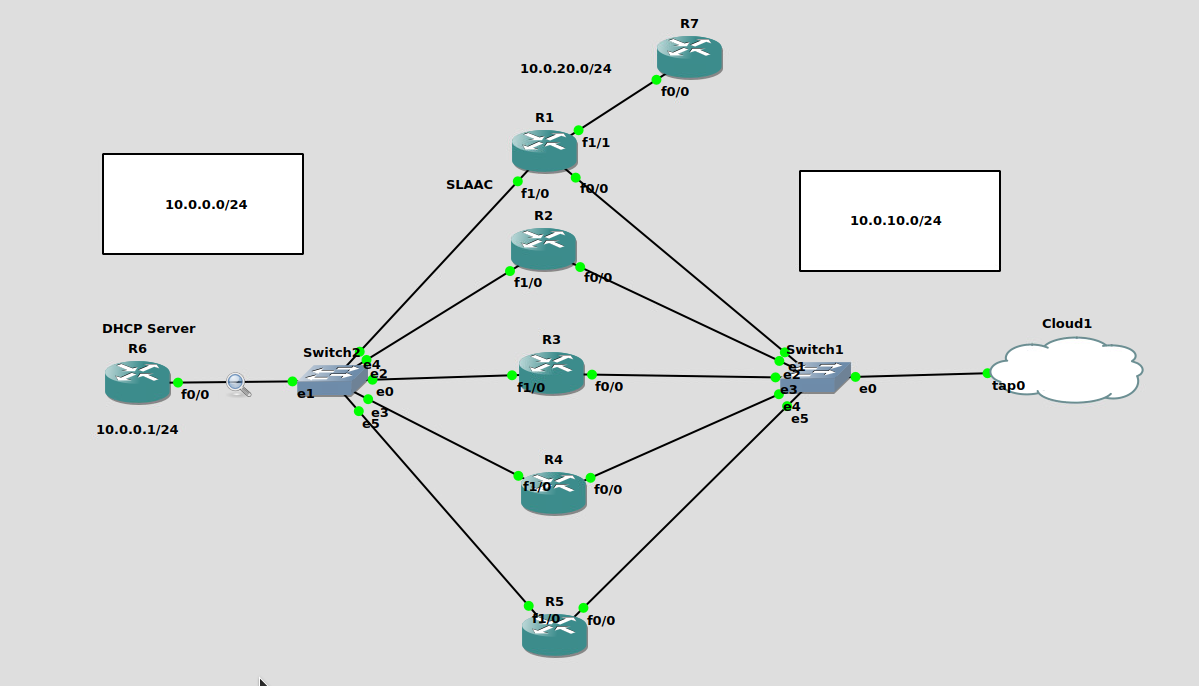




# PART 3: DHCP Relay [Extra Credit: 10 Points]

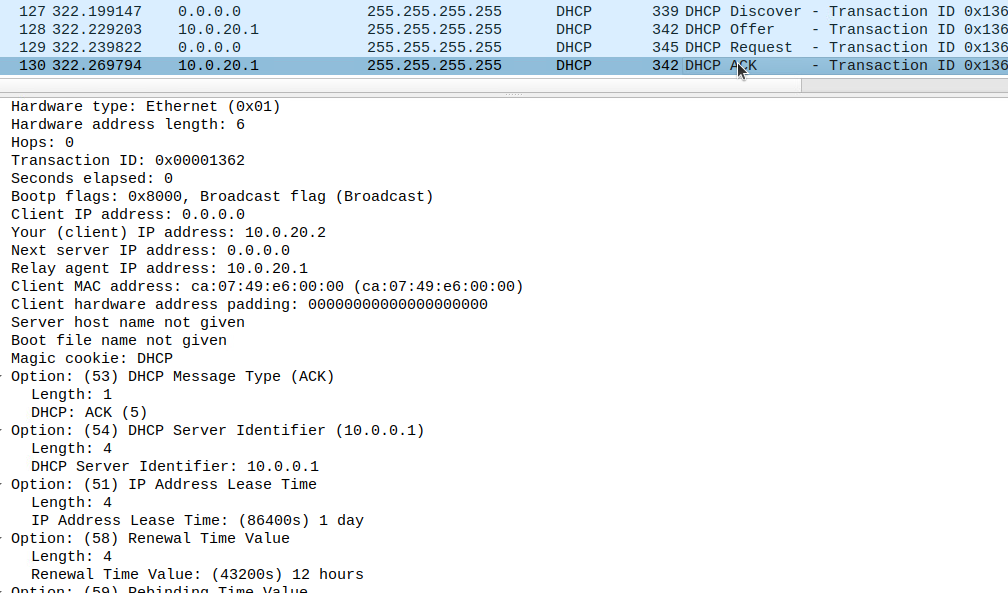
IPv4 DHCP Relay

1. Create a network design, where an additional router is added to the network (behind R1). Have this newly added router receive an IP address from the DHCP server, via the DHCP relay.

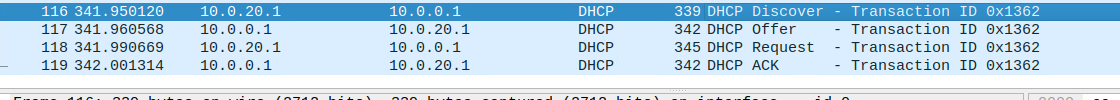


1. Provide a screenshot of the Wireshark capture showing the client receiving an address via DHCP relay.

Between R1 to R7 –

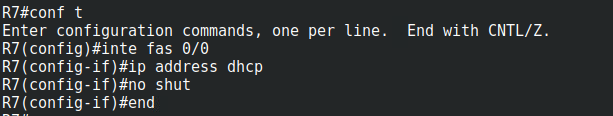


Between R6(DHCP Server) and R1(DHCP Relay) –

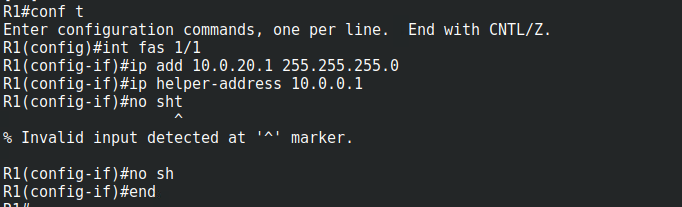


1. Provide the relevant router configurations that are required for DHCP relay.

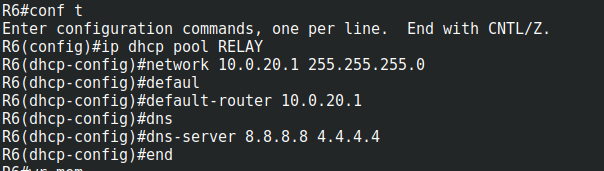
R7 (DHCP HOST) –



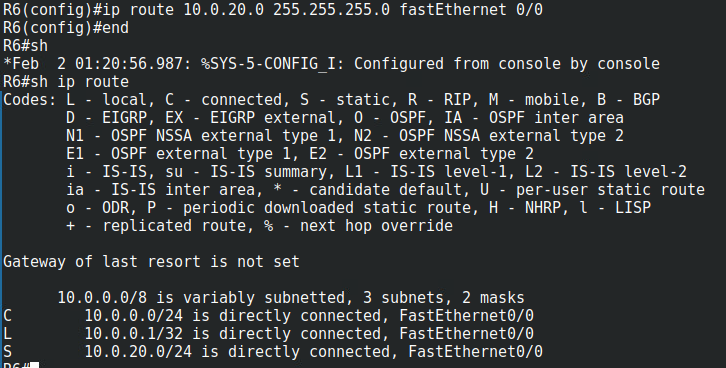
R1 ( IP Helper Address to Reach Router 1) -



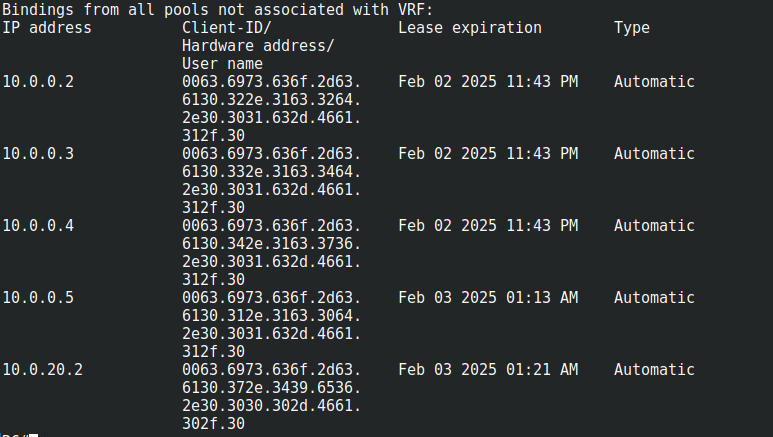
R6 (DHCP Server - pool for Relay)-



Reverse route for 10.0.20.0 network to reach to HOST -

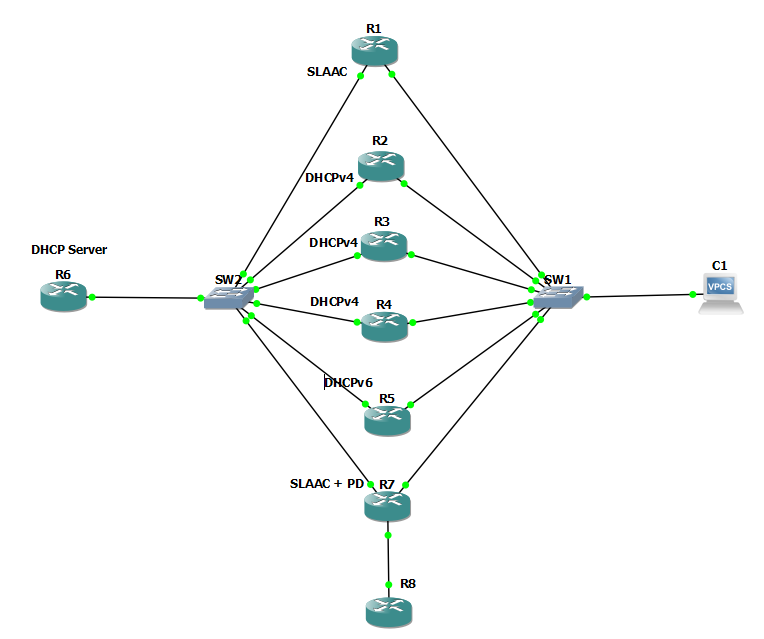


R7 DHCP Bindings -

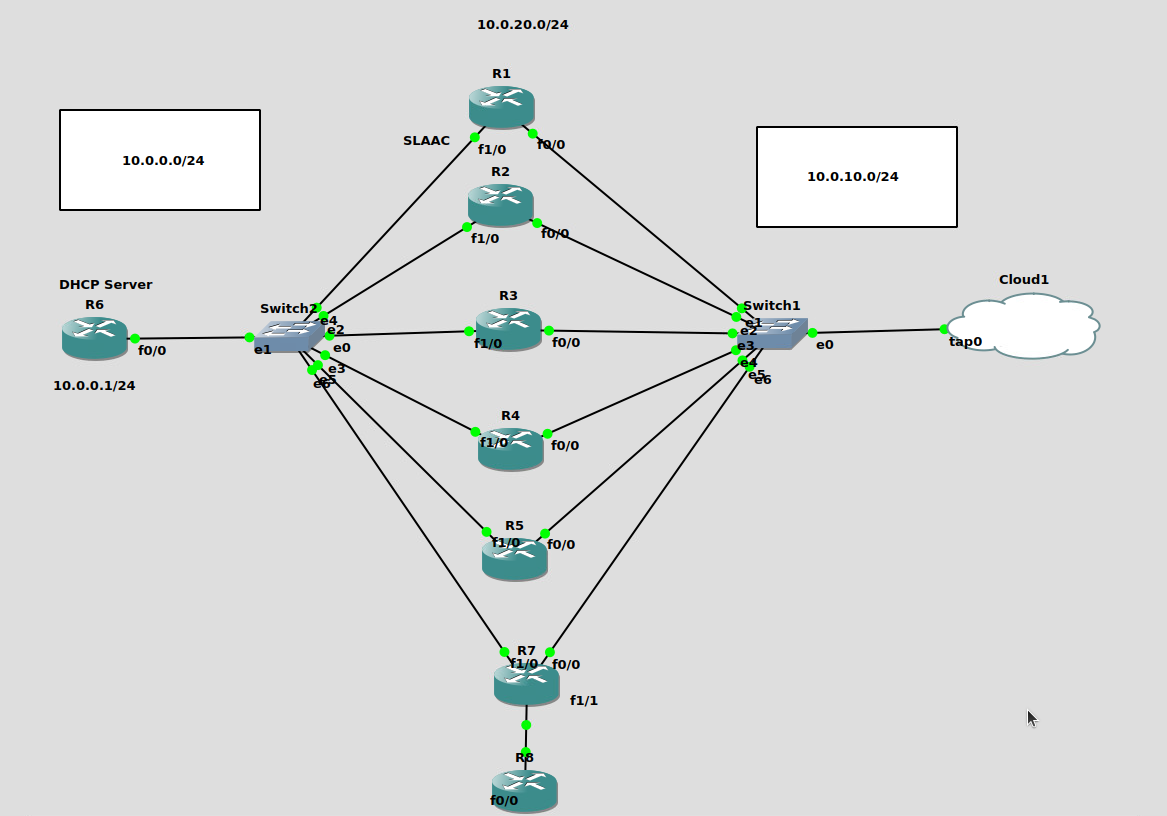


# PART 4: DHCPv6 Prefix Delegation [Extra Credit: 10 Points]

The aim of this objective is to make R6 allocate a prefix to R7 through DHCPv6 Prefix delegation and subsequently make R7 allocate an IPv6 address to R8 from the delegated prefix pool.

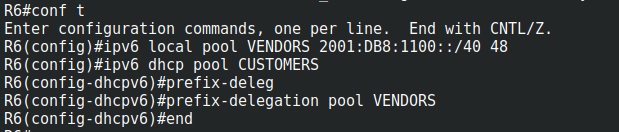


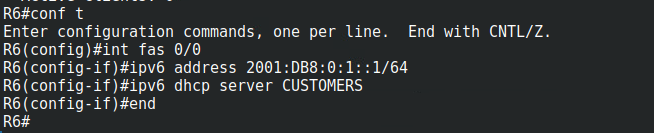
1. Create the topology as shown in the above figure.

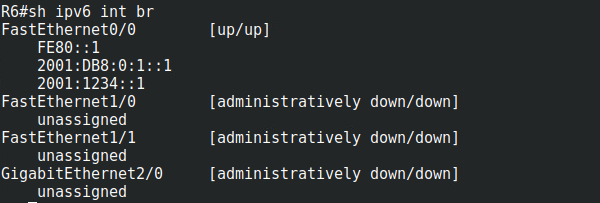


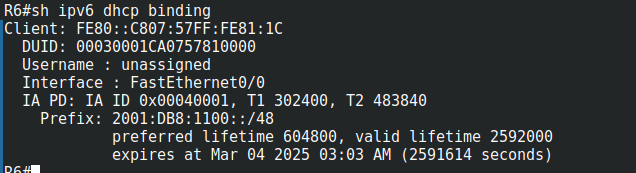
1. Configure R6 to act as a DHCPv6 Server to allocate a /48 Prefix to R7. In addition, the interface on R7 connecting to SW2 should receive an IPv6 address through SLAAC. Paste the screenshots of the relevant configuration on both R6 and R7.

R6 –

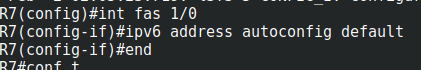


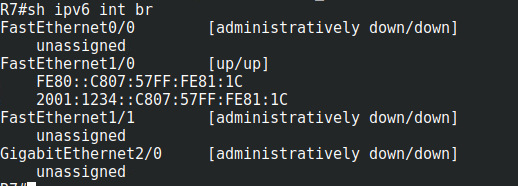






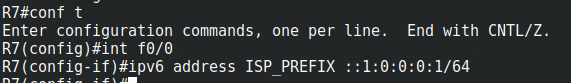
R7 –

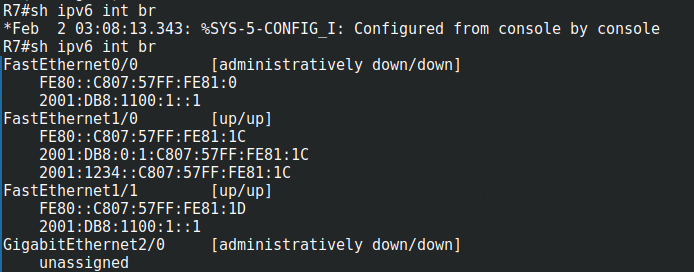




1. Configure R7 and R8 such that R8 receives a /64 address from the prefix pool delegated to R7 in the previous step. Paste the screenshots of the configuration on both R7 and R8.

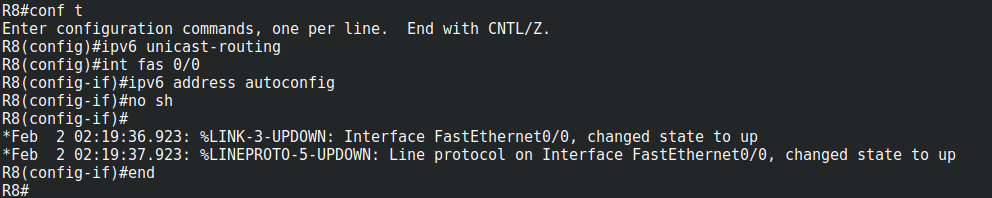
R7–

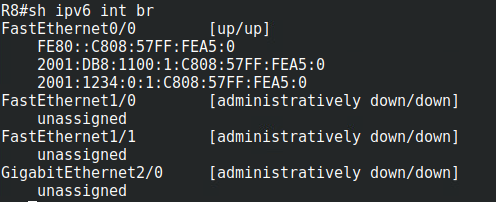






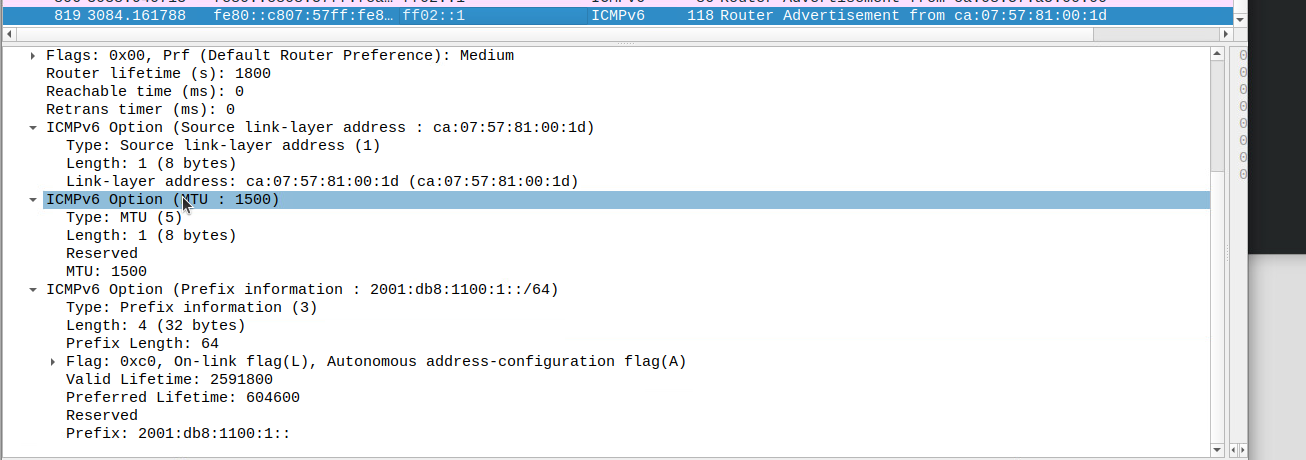
R8 –



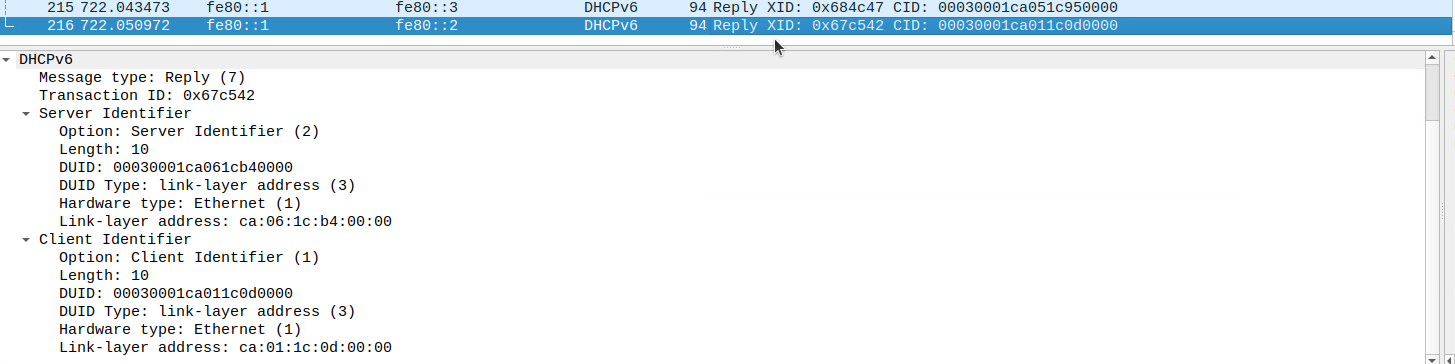


1. Provide screenshots of the Wireshark packet captures showing the prefix delegated to R7 and the address allocated to R8.

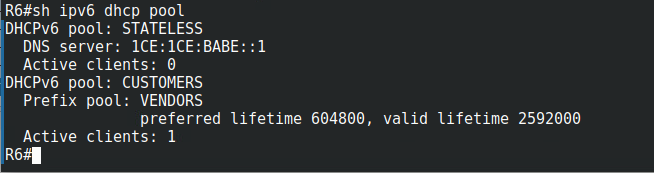
R8 –



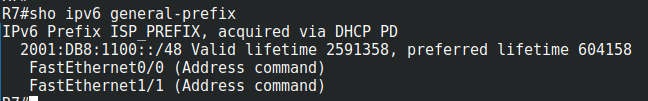
R7

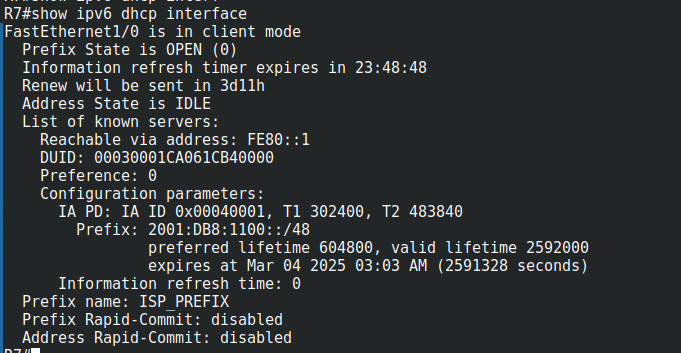


1. Provide a screenshot of the output for the following commands:
2. “show ipv6 dhcp pool” and “show ipv6 dhcp binding” on R6

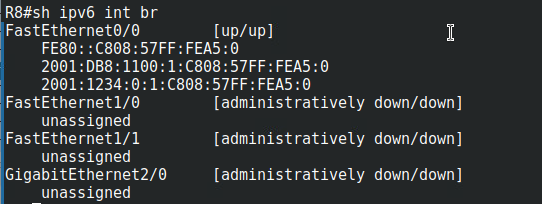


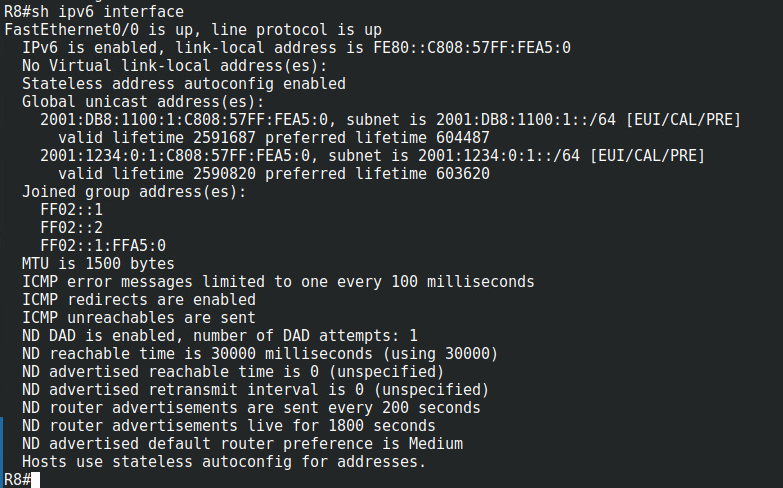
1. “show ipv6 general-prefix” and “show ipv6 dhcp interface” on R7





1. “show ipv6 interface brief” and “show ipv6 interface” on R8





**QUESTIONS: [25 Points]**

1. Explain the difference between stateless and stateful DHCPv6. **[5 points]**

**Stateless DHCPv6:**

* It provides the configuration parameters to hosts without assigning IP addresses
* It can offer configuration such as DNS server, domain names etc.

**Stateful DHCPv6:-**

* It is used to assign the IP address to hosts along with other parameters
* Host rely on these servers for the IP assignments and configuration details.

1. When will you use an IPv6 helper address in a DHCPv6 configuration? Explain briefly how and why you would use an IP helper address. **[5 points]**

The helper address is used when DHCP server is deployed in different network so DHCP broadcast can only reach gateway and there we need to configure the helper address on the interface to send the DHCP packets to DHCP server.

1. Explain the concept of prefix delegation? [ Hints: Delegating Router and Requesting Router] **[5 points]**

When a router assigns a portion of its IPv6 address space to another router or network on demand is called prefix delegation.

The delegating router allocates a prefix by carving out a space from its pool which allows it to configure devices within its network segment.

1. Explain the significance of DUID? **[5 points]**

DUID is unique identifier provided by servers to client which acts as a stable and persistent identifier for clients while ensuring uniform configuration and better management of address space.

1. Briefly describe how DHCPv6 prefix delegation can induce security risks in the network? State ways to mitigate these risks. **[5 points]**

Security Risk –

Attackers can create unauthorized subnets, intercept traffic via man-in-the-middle attacks, overload resources with excessive prefix requests, and allow unauthorized devices to use the network.

**Mitigation Strategies:**

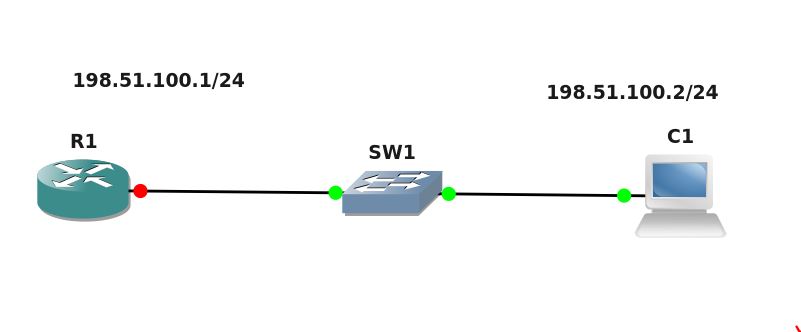
Limit prefix requests, use authentication, monitor DHCPv6 logs for suspicious activity, and filter unauthorized prefixes.

# SCAPY [95 Points]

Hackers are constantly adapting to network security techniques. Our network security ingenuity needs to improve and adapt to be able to protect dynamic security attacks. Scapy is one of the tools that gives us the ability to create, forge, and decode our own packets, send them on the network, capture them, and much more. Imagine how useful this tool can be to emulate and study the attacks, analyze the network behavior, and help in developing techniques to keep our networks safe and secure.

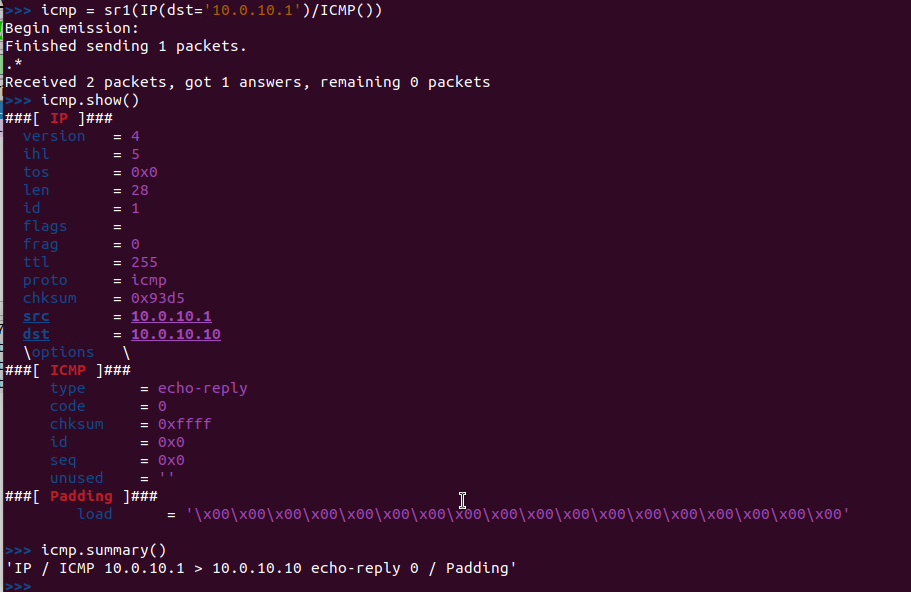
Pre-requisites: -

* Before proceeding with the Scapy objectives, create the topology shown below.
* Use the Scapy documentation to get started - <https://scapy.readthedocs.io/en/latest/>

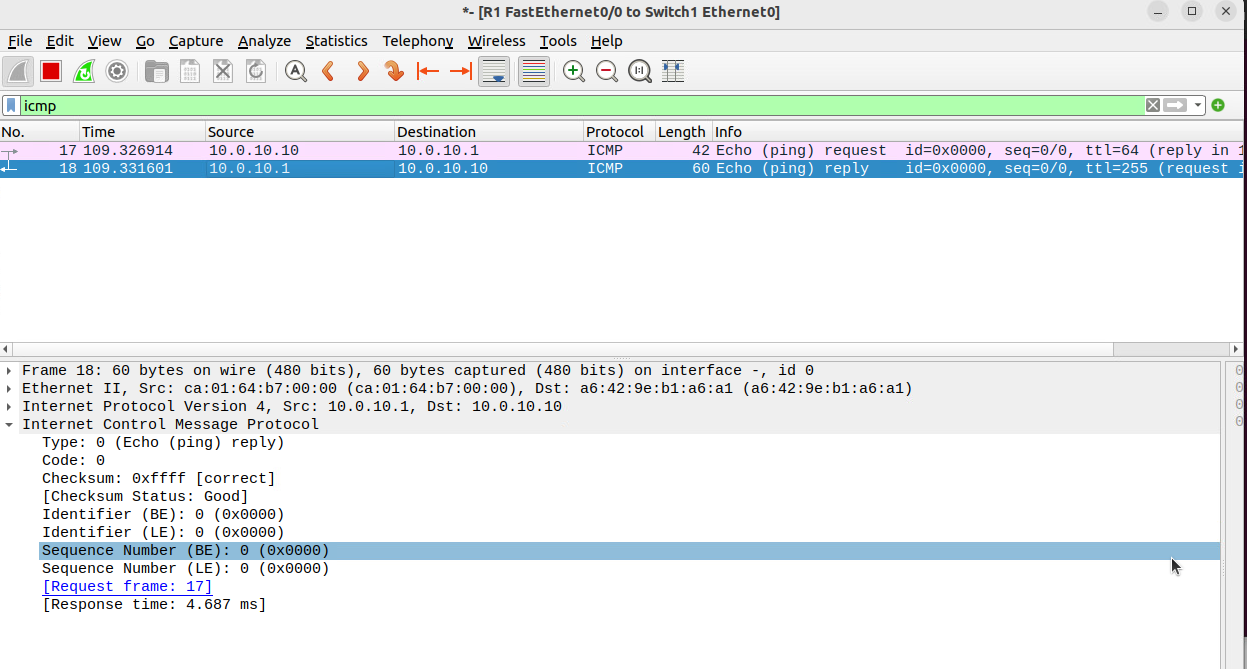


Objective 1: Creating an ICMP echo request packet and ARP frame through Scapy:

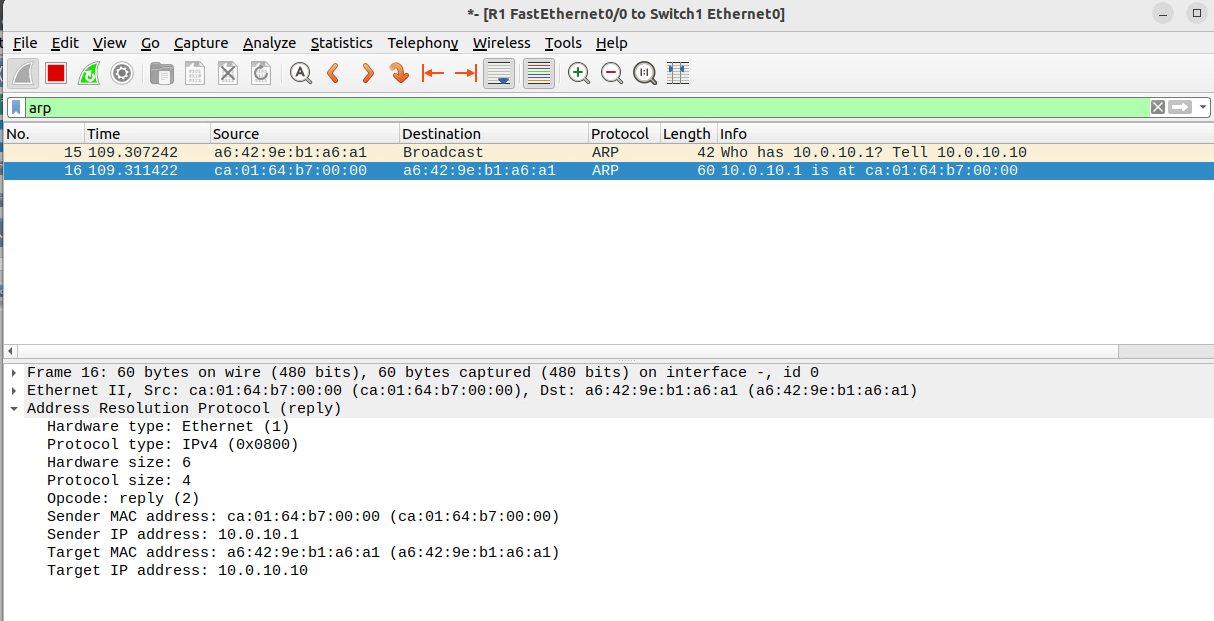
1. Create your own ICMP echo request packet on the VM (C1) to ping R1’s interface using Scapy. Show the packet structure in Scapy using appropriate show commands. **[10 points]**



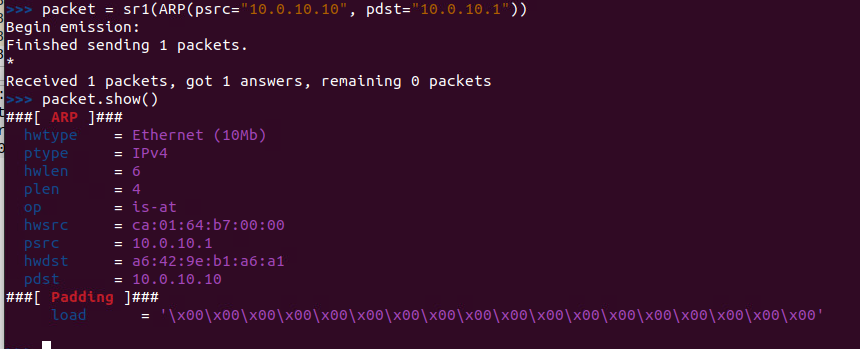
1. Before starting this objective, start a Wireshark capture on the correct interface in the above topology. Now send your Scapy generated echo request packet to router R1. Did you get a response for the above echo request? If yes, show the Wireshark capture indicating the response as well as the packet **received** in Scapy. If no, check if your echo request packet has the correct fields and it is being successfully delivered to R1. Try again after troubleshooting and resolving your issue to complete the objective. **[10 points].**



1. Using the Wireshark capture from step 2, filter the ARP exchange messages. Paste relevant screenshots. If you do not see ARP messages from the previous objective in Wireshark, create a new topology as above in GNS3, start Wireshark capture and then initiate a ping to R1 from C1’s terminal. Filter the ARP exchange and show appropriate screenshots. Good understanding of ARP exchange messages and frame format will help you in the next objective. **[10 points]**

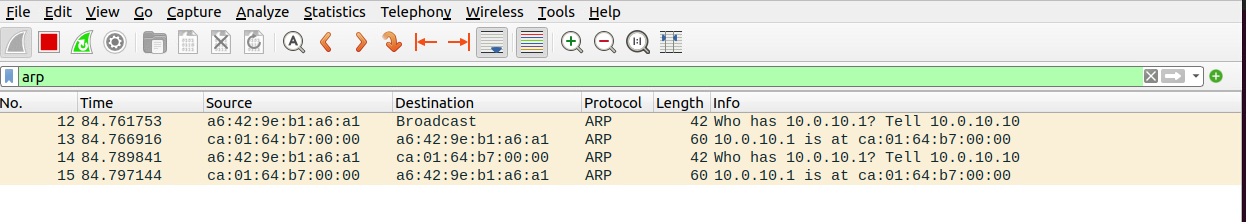


1. Start your Wireshark capture in the above topology before starting this objective. Recreate an ARP request destined to router R1 in Scapy and capture the ARP response in Wireshark to find the MAC address of the router. Paste relevant screenshots indicating successful ARP response for your Scapy generated ARP request. **[10 points]**

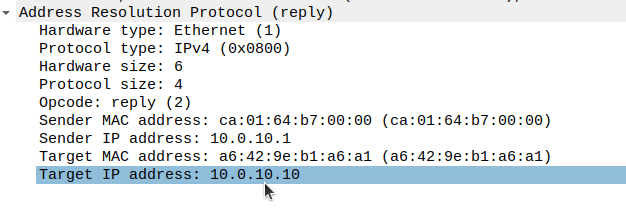


1. Based on your understanding of ARP, answer the following questions **[5 points]:**

* What filter did you use to only display ARP messages and what are the contents of an ARP message?



Type arp and we can see all the packets filtered by specific protocol.



The ARP contains following fields –

OP Code, Sender mac, receiver mac, sender IP and receiver IP

* Which field in an ARP message is used to identify if it is an ARP request or an ARP reply?

OPCode is used to identify if its request and reply.

1 for request and 2 for reply.

* What is a Gratuitous ARP and why is it used?

**Gratuitous ARP (GARP)** is an ARP request or reply sent by a device to announce or update its IP-to-MAC mapping.

It is used for detecting IP conflicts, updating ARP caches, and informing network devices of a MAC address change.

* Which layer in the OSI model does ARP belong and why?

ARP Operates at Data-Link Layer which is layer .

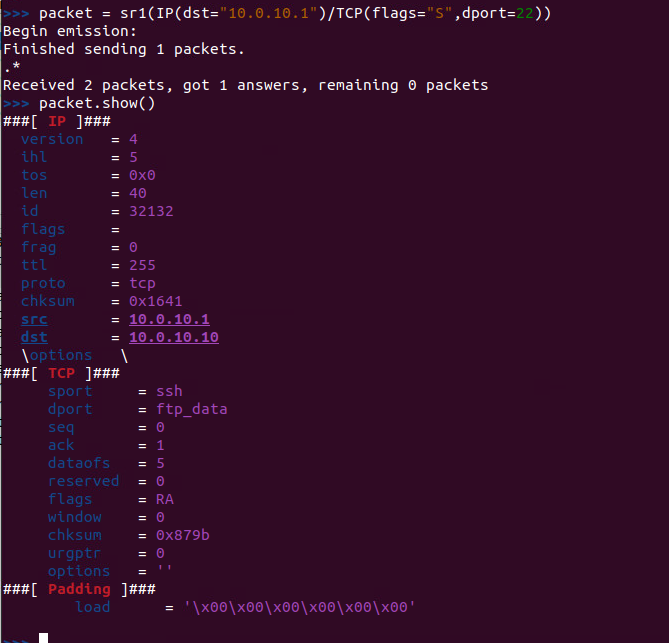
* What is Proxy ARP and why is it used?

**Proxy ARP** allows a router to respond to ARP requests on behalf of another device by providing its own MAC address, enabling communication between hosts that believe they are on the same subnet but are separated by a router.

Objective 2: SYN flood using Scapy:

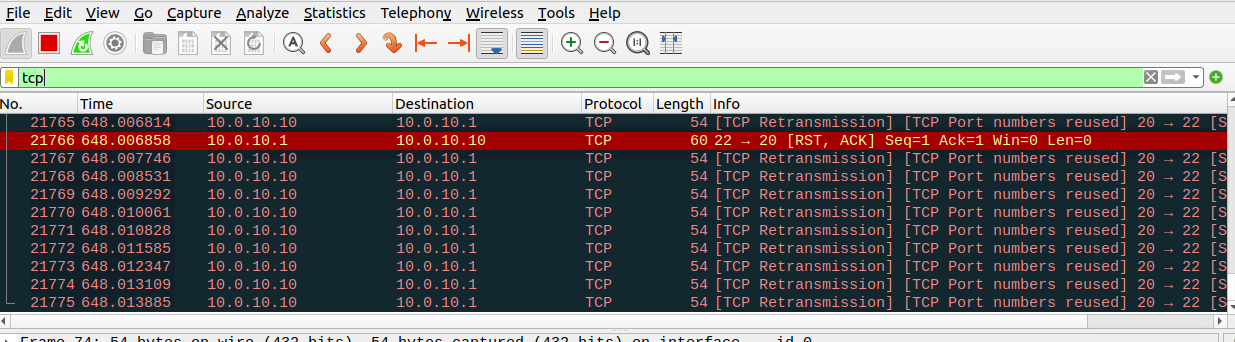
“Bringing down a server” is typically a moment of pride in the world of hackers. SYN flood is one of the popular denial-of-service attacks that targets the end system (especially a server). It is an event where the attacker sends a succession of SYN requests (to which the server responds with a SYN-ACK) with an intent to consume enough resources that will make the server unresponsive to legitimate traffic and eventually bring it down. In this objective you will create a SYN packet and initiate a SYN flood attack using Scapy.

1. Create your own SYN packet using Scapy and display the packet structure. **[10 points]**

****

1. Initiate a SYN flood attack to the router’s interface from the VM (C1). Show how you initiated the attack through Scapy and show the relevant traffic on Wireshark. **[20 points]**



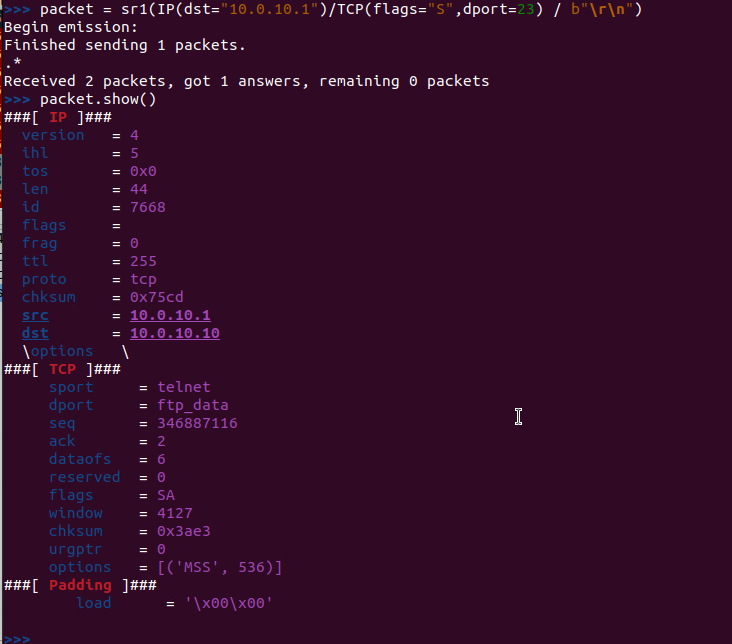


1. Give two ways on how you will prevent such an attack from happening. **[5 points]**

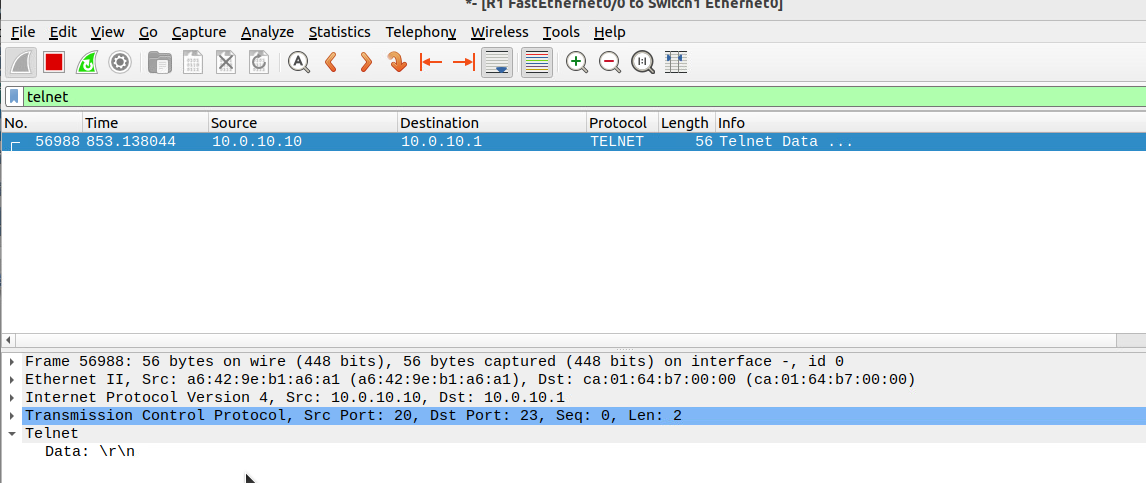
* We can implement next generation firewall which can isolate this kind of issue and block these type of packets
* Reducing the keep-alive timer for TCP session which will allow for resources to free up faster avoiding these kind of issues.

Objective 3: Creating your own Telnet packet using Scapy:

1. Create your own Telnet packet using Scapy and display the packet structure. **[10 points]**

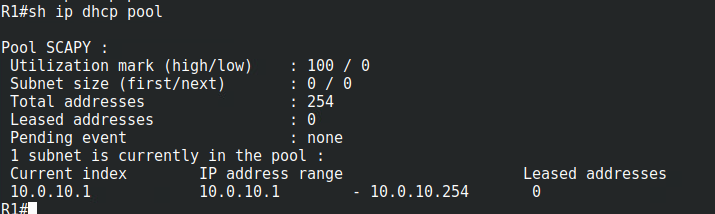
****

1. Display the Telnet traffic using Wireshark. **[5 points]**

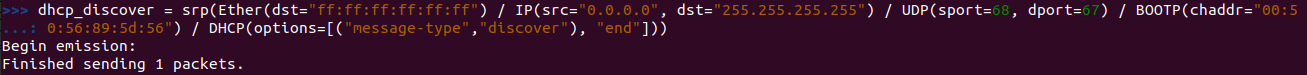


Extra Credit 1 [5 points]

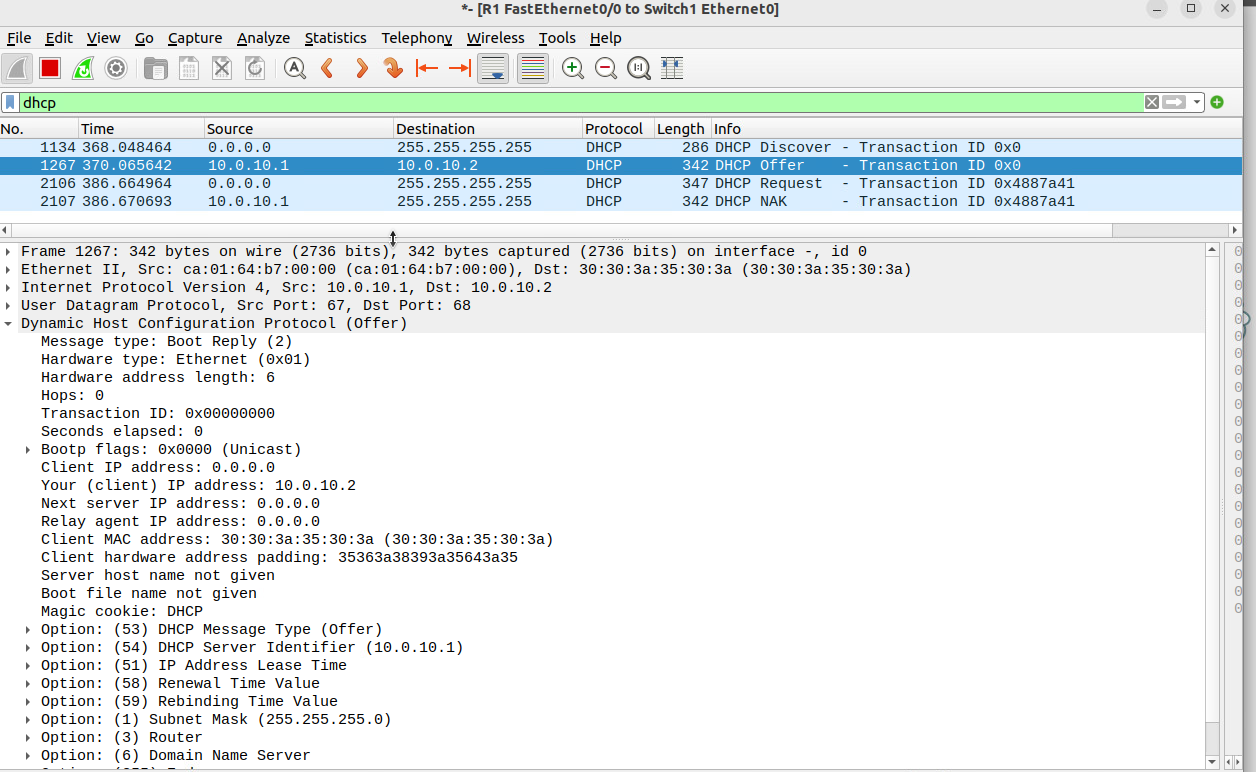
Make R1 as the DHCP server for your VM (C1) as the client.

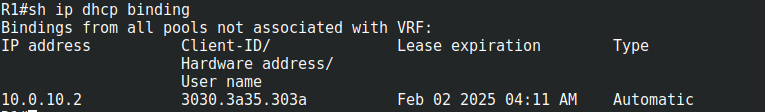


1. Create your own DHCP discover packet using Scapy and display the packet structure.

****

1. Show the DHCP Server reply packets on Wireshark.

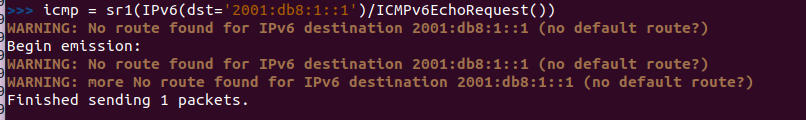
****

****

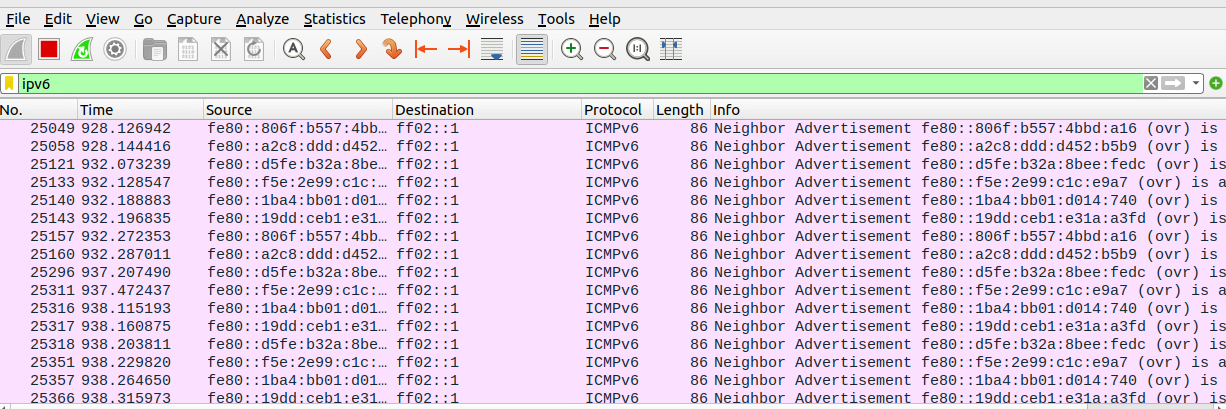
Extra Credit 2 [5 points]

1. Repeat the Scapy Objective 1 (Create an ICMP echo request and ARP frame through Scapy) using IPv6. Answer questions 1 to 4 with respect to IPv6. Paste relevant screenshots. (HINT: Are you sure it is ARP?)

ICMP Packet using IPV6 in SCAPY



I added a route for IPV6 traffic to go to TAP interface



Reflection:

What did you learn from this lab and how would you incorporate this in a production network setting? **[5 points]**

**I learned DHCPv4, DHCPv6 autoconfiguration, prefix-delegation and how stateless DHCP configuration can be used to share the attributes.**

**I also learned to how I can utilize scapy to validate the configuration even when we do not have a client available for testing and we can craft corner case packets to validate the behavior of the network.**

**I would use this this lab learning in my day to day use on production setting like validating the state of ports or firewall configuration using scapy or I will use programmatically way to configure devices at scale**

# Total Score = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/190 + (30 Bonus)