Lab2

IP Assignment and Hop-by-hop Forwarding

TA: Yen-An Chien

Deadline: 2022/03/14 (MON) 23:59

Outline

- Objective
- Introduction to DHCP
- Lab environment
- Lab requirement
- Appendix

Outline

- Objective
- Introduction to DHCP
- Lab environment
- Lab requirement
- Appendix

Objective

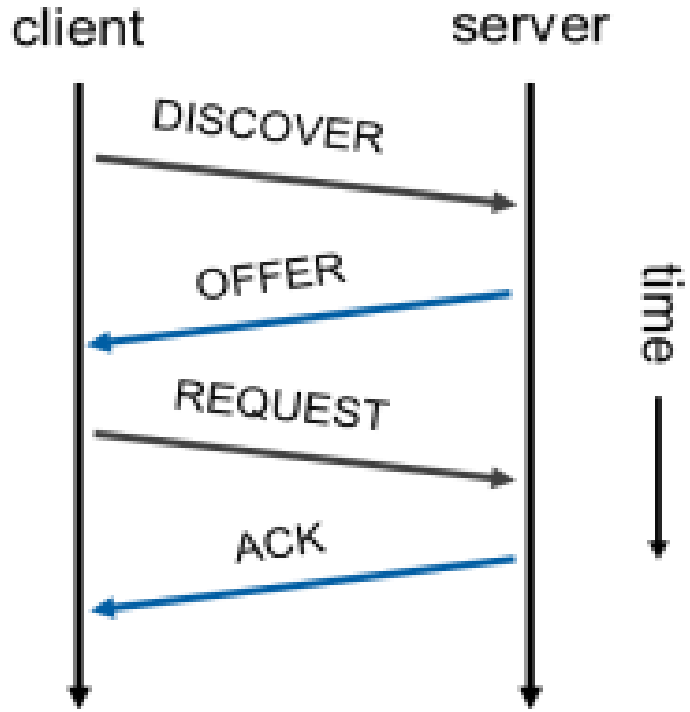
- Subnetting and Netmask
- Static Routing Rule Configuration
- DHCP Server configuration
- DHCP 4-way Handshaking Messages
- Traceroute Observation

Outline

- Objective
- Introduction to DHCP
  - What is DHCP
- Lab environment
- Lab requirement
- Appendix

Dynamic Host Configuration Protocol (DHCP)

- Provide necessary information for a host to access network
  - IP address, Gateway, DNS (Domain Name Server), etc
- Client and server use UDP port 68 and 67, respectively
- A DHCP transaction consists of 4 messages



Outline

- Objective
- Introduction to DHCP
- Lab environment
  - Environment Setting and DHCP Utilities
  - Lab topology
  - Python script for lab topology
- Lab requirement
- Appendix

Environment and Utility Installation

- Install Ubuntu, mininet and Wireshark as in Lab 1
- Install DHCP Server and Client

```
bash$ sudo apt update && sudo apt upgrade -y
bash$ sudo apt install isc-dhcp-server -y
bash$ sudo apt install isc-dhcp-client -y
```

- Install traceroute
  - Install traceroute to trace hops details of routing paths

```
bash$ sudo apt install traceroute -y
```

Enabling DHCP Server and Client

- AppArmor
  - Linux application security system.
  - Proactively protects operating system and application
- Modify AppArmor settings **(done only for the first time)**
  - For server

```
bash$ sudo ln -s /etc/apparmor.d/usr.sbin.dhcpd \
                 /etc/apparmor.d/disable/
bash$ sudo apparmor_parser -R /etc/apparmor.d/usr.sbin.dhcpd
```

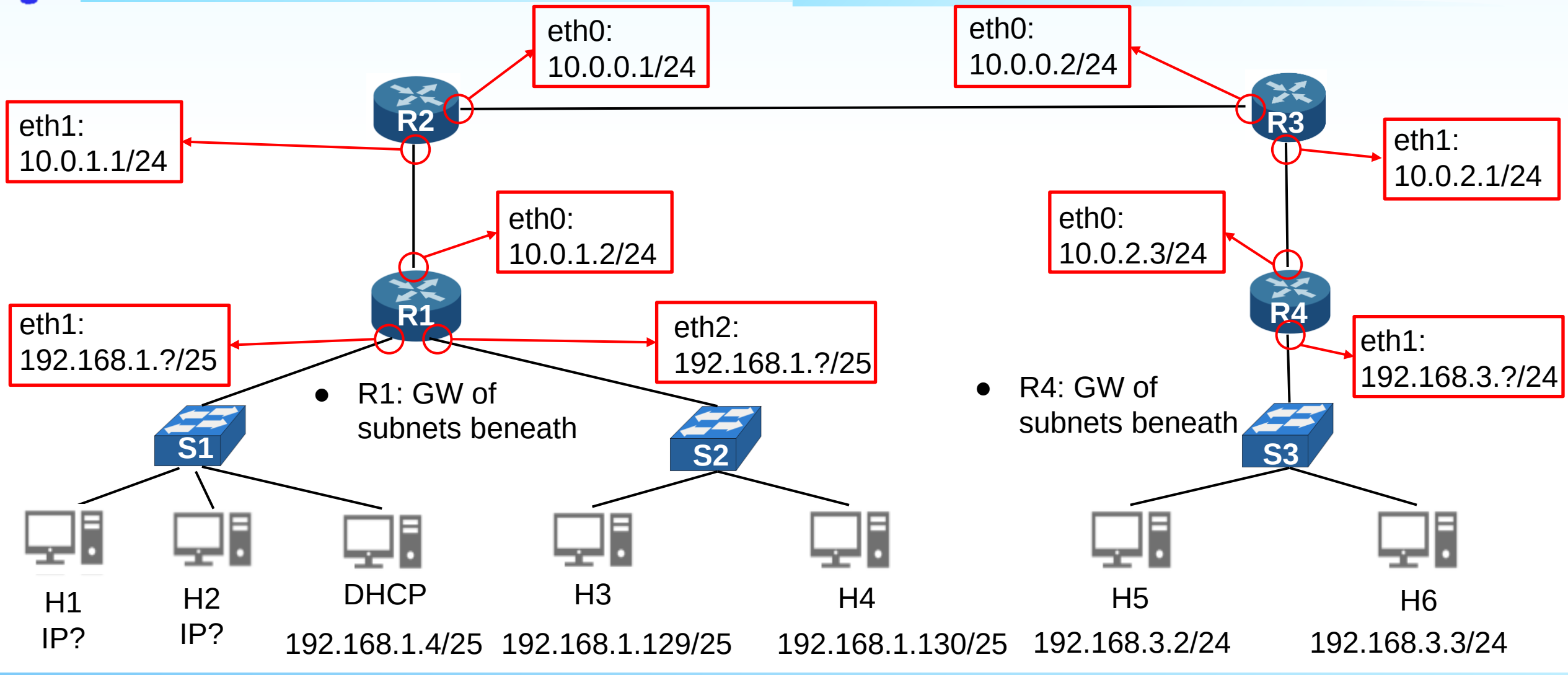
  - For client

```
bash$ sudo /etc/init.d/apparmor stop
bash$ sudo sed -i '30i /var/lib/dhcp{,3}/dhcpclient* lrw,' \
                       /etc/apparmor.d/sbin.dhclient
bash$ sudo /etc/init.d/apparmor start
```

Outline

- Objective
- Introduction to DHCP
- Lab environment
  - Environment Setting and DHCP Utilities
  - Lab topology
  - Python script for lab topology
- Lab requirement
- Appendix

Lab Topology



Outline

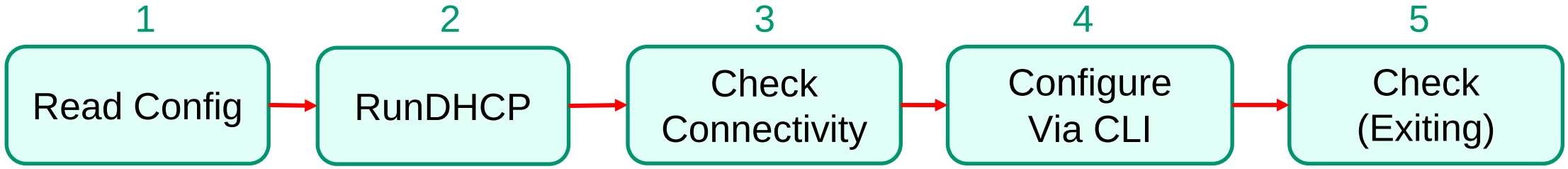
- Objective
- Introduction to DHCP
- Lab environment
  - Environment Setting and DHCP Utilities
  - Lab topology
  - Python script for lab topology
- Lab requirement
- Appendix

Python Script for Lab Topology

- Topology.py: a Python script for lab topology
  - Download from E3
- Create and put dhcpd.conf at the same directory as topology.py
  - dhcpd.conf: configuration file for DHCP daemon
- Run topology.py to create the topology

```
bash$ sudo python topology.py
```

- Components and Sequence of topology.py



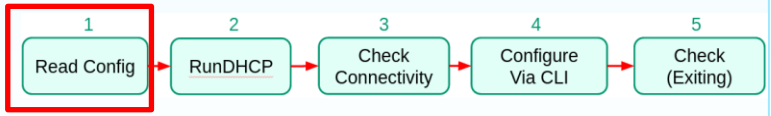
DHCP Configuration file – dhcpd.conf

- Create dhcpd.conf in the same directory as topology.py
- dhcpd.conf

```
subnet [subnet] netmask [netmask] {
    range [begin] [end];
    option routers [gateway IP];
    option subnet-mask  [subnet-mask];
}
```

topology.py: 1 – Read config

- config(): node configuration script (marked and incompleted)
  - Configure IPs and Default gateways for hosts
  - Configure IPs and Static Routes for routers

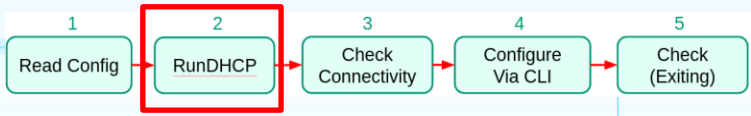



```
78  def config(hosts, switches, routers, DHCPServer):
79      # Hosts interface IP and  default gateway configuration
80      DHCPServer.cmd('ifconfig DHCPServer-eth0 [IP/prefix]')
81      hosts['h2'].cmd('ifconfig h2-eth0 [IP/prefix]')
82      hosts['h2'].cmd('route add default gw [gatewayIP]')
83      # ...
84      #Routers interface IP configuration
85      routers['r1'].cmd('ifconfig r1-eth0 [IP/prefix]')
86      # ...
87      # Router routing table configuration
88      routers['r1'].cmd('route add -net [networkID/prefix] gw [peer IP]')
89      # ...
```

Prefix Length

Add a static route to a network

topology.py: 2 – RunDHCP



- **runDHCP()**
  - Run DHCP server in mininet node (DHCPServer)

```
64      # Run DHCP server at node DHCPserver
65      #runDHCP(net) # if your dhcpd.conf is done, uncomment this line
129 def runDHCP():
130     #Run DHCP server on node DHCPServer
131     print("[+] Run DHCP server")
132     dhcp = net.getNodeByName('DHCPServer')
133     dhcp.cmdPrint('/usr/sbin/dhcpd 4 -pf /run/dhcp-server-dhcpd.pid \
            -cf ./dhcpd.conf %s' % dhcp.defaultIntf())
```

IPv4

Store PID of dhcpd for DHCPServer

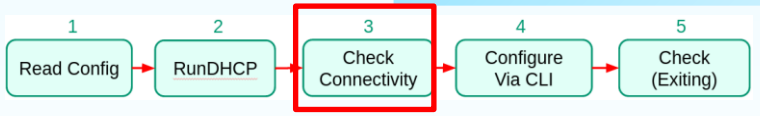
Run this daemon (dhcpd)

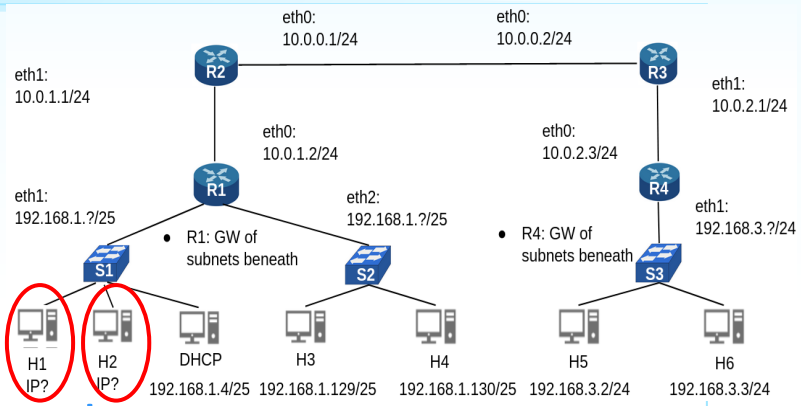
Use this config (dhcpd.conf)

Run dhcpd on this Interface of DHCPServer

topology.py: 3 – Check Connectivity

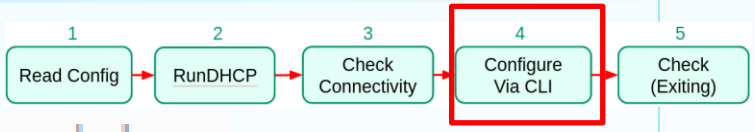
- check()
  - Script that checks the correctness of your configuration until now
  - Recall: h1 and h2 does not have IP yet
  - All hosts except h1 and h2 should be able to ping one another
  - Check starts from h1 to other hosts, then the next to the remaining hosts
  - Print WORNG ANSWER if fails






```
anthony@yachien-ubuntu:~/Lab2_v0$ sudo python TA_topology.py
[sudo] password for anthony:
[+] Run DHCP server
h1 doesn't have connectivity to 192.168.1.129
h1 doesn't have connectivity to 192.168.1.130
h1 doesn't have connectivity to 192.168.3.2
h1 doesn't have connectivity to 192.168.3.3
h2 doesn't have connectivity to 192.168.1.129
h2 doesn't have connectivity to 192.168.1.130
h2 doesn't have connectivity to 192.168.3.2
h2 doesn't have connectivity to 192.168.3.3
WRONG ANSWER
mininet>
```

topology.py: 4 – Configure Via CLI



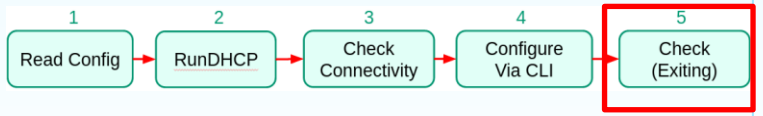
- Launch mininet CLI

```
70      # Comment this line if you don't need to debug
71      CLI(net)
```

  - to Debug
    - ping hosts, traceroute, …
  - To perform more configuration
    - add routing rules, change IPs, …
- E.g., Configure IP and Gateway of h1
  - Run DHCP Client on h1 with eth0

```
mininet> h1 dhclient h1-eth0
```

topology.py: 5 – Check (Exiting)



- check()
  - Before exit mininet, topology.py will perform check() again

```
mininet> exit
ACCEPT
[-] Killing DHCP server
anthony@yachien-ubuntu:~/Lab2_v0$
```

  - All hosts should now reach one another
    - Print ACCEPT

Outline

- Objective
- Introduction to DHCP
- Lab environment
- Lab requirement
  - Part1: Complete topology.py (20%)
  - Part2: DHCP Server configuration (25%)
  - Part3: Answer Questions (20%)
- Appendix

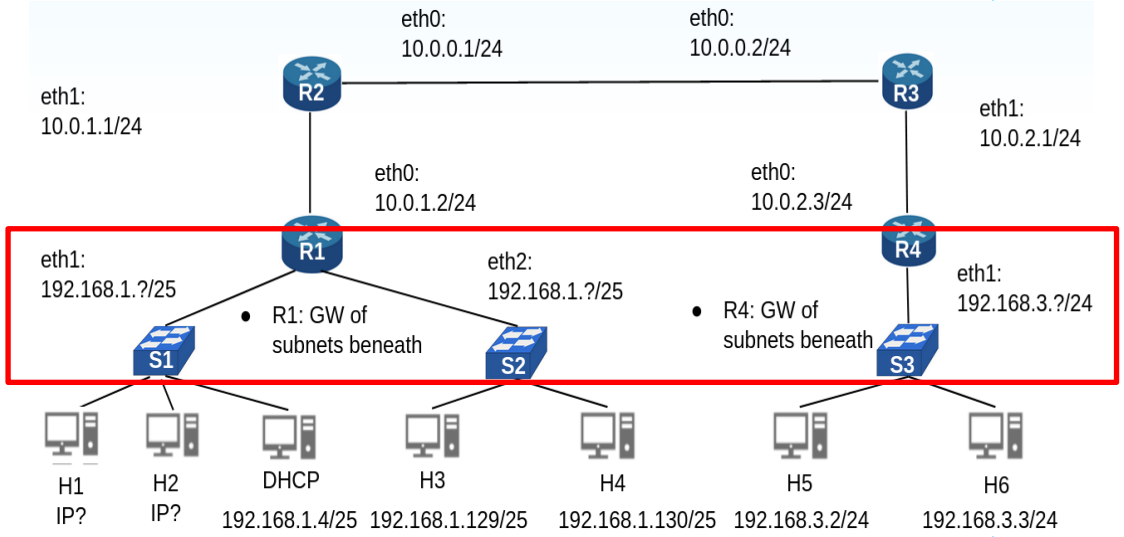
Step 1: Complete topology.py

- Complete config() function to configure nodes

Step 1-1. Set IP address of Hosts

Step 1-2. Configure Routers and default gateway for hosts

  - Set IP address on all interfaces of Routers
    - Gateway address of a subnet must be the second last address of the subnet
  - Set static routing rules
    - Add a static route to each network



Part 1 Questions

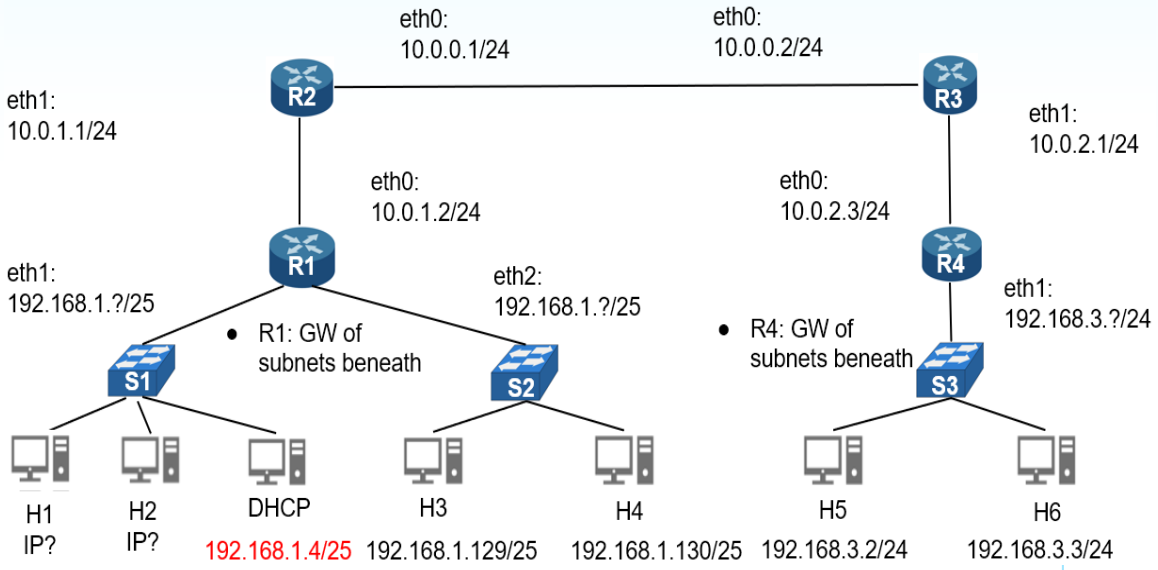
1. After you complete Steps 1-1
   a) Can h3 ping h4? Briefly explain why or why not.(5%)
   b) Can h3 ping h5? Briefly explain why or why not.(5%)

Complete topology.py so that all hosts, except h1 and h2, can ping one another.

2. Take screenshot to show that your topology configuration is correct. (10%)

Step 2: DHCP server configuration

- Create and put a dhcpd.conf at the same directory
  - Configuration parameters
    - IPs pool of 192.168.1.0/25
    - Default gateway for 192.168.1.0/25
- h1: Dynamic IP assignment
  - DHCP randomly assigns an IP to h1
- h2: Fixed IP assignment
  - DHCP assigns a fixed IP to h2 (according to it's mac address)
    - The last two digits of the IP address must be the same as the last two digit of your student ID
      - If the last two digits of your ID is the same as the ones of DHCP
        - Choose two arbitrary digits (You need to explain your choice)



Part 2 Questions

3. Run dhcp on h1 and capture DHCP messages. Take screenshot to show the IPs and MACs. (10%)

```
mininet> h1 wireshark &   #listen at h1-eth0
mininet> h1 dhclient h1-eth0 #
```

4. Can hosts other than h1 and h2 acquire IP addresses from DHCP server? Briefly explain your answer. (5%)

Part 2 Questions (Cont'd)

5. Run dhcp on h2 and take screenshot to show IP of h2. (5%)

```
anthony@yachien-ubuntu:~/Lab2_v0$ sudo python TA_topology.py
[sudo] password for anthony:
[+] Run DHCP server
h1 doesn't have connectivity to 192.168.1.129
h1 doesn't have connectivity to 192.168.1.130
h1 doesn't have connectivity to 192.168.3.2
h1 doesn't have connectivity to 192.168.3.3
h2 doesn't have connectivity to 192.168.1.129
h2 doesn't have connectivity to 192.168.1.130
h2 doesn't have connectivity to 192.168.3.2
h2 doesn't have connectivity to 192.168.3.3
WRONG ANSWER
mininet> h1 dhclient h1-eth0
mininet> h2 dhclient h2-eth0
mininet> h2 ifconfig h2-eth0
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.65  netmask 255.255.255.128  broadcast 192.168.1.127
        inet6 fe80::200:ff:fe00:2  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:00:00:02  txqueuelen 1000  (Ethernet)
        RX packets 49  bytes 5468 (5.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 1198 (1.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet> exit
ACCEPT
[-] Killing DHCP server
anthony@yachien-ubuntu:~/Lab2_v0$
```

6. Explain how you make dhcp server assigned fixed address to h2. (5%)

Part 3 Questions (1/3)

- Invoke wireshark on node r1 and answer questions

```
mininet> r1 wireshark &   #listen at r1-eth0
mininet> r1 wireshark &   #listen at r1-eth1
mininet> h1 ping h5 –c 1
```

7. What does r1 do on the packets from h1 to h5, and h5 to h1, respectively? Capture packets and show screenshot to explain your answers. (5%)

Part 3 Questions (2/3)

- Activate Wireshark on all routers' eth0, and execute traceroute on h1

```
mininet> [r1-r4] wireshark &   #listen at [r1-r4]-eth0
mininet> h1 traceroute h5
```

8. Capture packets and take screenshot to answer the following two questions.

a) Show the first six ICMP Unreachable messages. (5%)

Ultimately, h1 will send traceroute packets to h5 successfully.

b) Show the ICMP packets forwarded by each router, which constitute the first successful delivery of a traceroute packet from h1 to h5. (5%)

Part 3 Questions (3/3)

- Ideally, we should have all the hop details as follows.

```
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.62)  0.283 ms  0.015 ms  0.006 ms
 2  10.0.1.1 (10.0.1.1)  0.017 ms  0.008 ms  0.008 ms
 3  10.0.0.2 (10.0.0.2)  0.016 ms  0.009 ms  0.012 ms
 4  10.0.2.3 (10.0.2.3)  0.017 ms  0.011 ms  0.018 ms
 5  192.168.3.2 (192.168.3.2)  0.193 ms  0.026 ms  0.027 ms
```

9. Briefly explain why sometimes we may have hop details with "***." Take Wireshark screenshots to justify your answer. (5%)

```
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.62)  0.349 ms  0.255 ms  0.194 ms
 2  10.0.1.1 (10.0.1.1)  0.247 ms  0.260 ms  0.242 ms
 3  * * *
 4  * * *
 5  192.168.3.2 (192.168.3.2)  0.186 ms  0.180 ms  0.170 ms
```

Report Submission

- Files
  - <studentID>_topology.py (20%)
  - dhcpd.conf (15%)
  - A report: lab2_<studentID>.pdf (65%)
    - Part1, Part2, Part3 Question Answers
- Submission
  - Zip python script, dhcpd.conf and the report into a zip file
    - Name: lab2_<studentID>.zip
  - Wrong file name and format subjects to 10 points deduction

Outline

- Objective
- Introduction to DHCP
- Lab environment
- Lab requirement
- Appendix

Appendix

- route table basic usage (mininet> [node] [command])
  - Check current routing rules

```
bash$ route
```

  - Add default gateway on a host

```
bash$ route add default gw [gateway IP]
```

  - Add routing rules on router

```
bash$ route add -net [subnet] gw [gateway IP]
```

- Change IP address of an interface

```
bash$ ifconfig [interface] [IP]
```

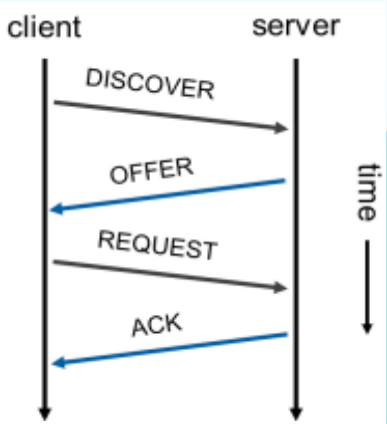
- Show all interfaces

```
bash$ ifconfig
```

Appendix

- AppArmor main page
  - https://ubuntu.com/server/docs/security-apparmor
- dhcpd.conf main page
  - https://linux.die.net/man/5/dhcpd.conf

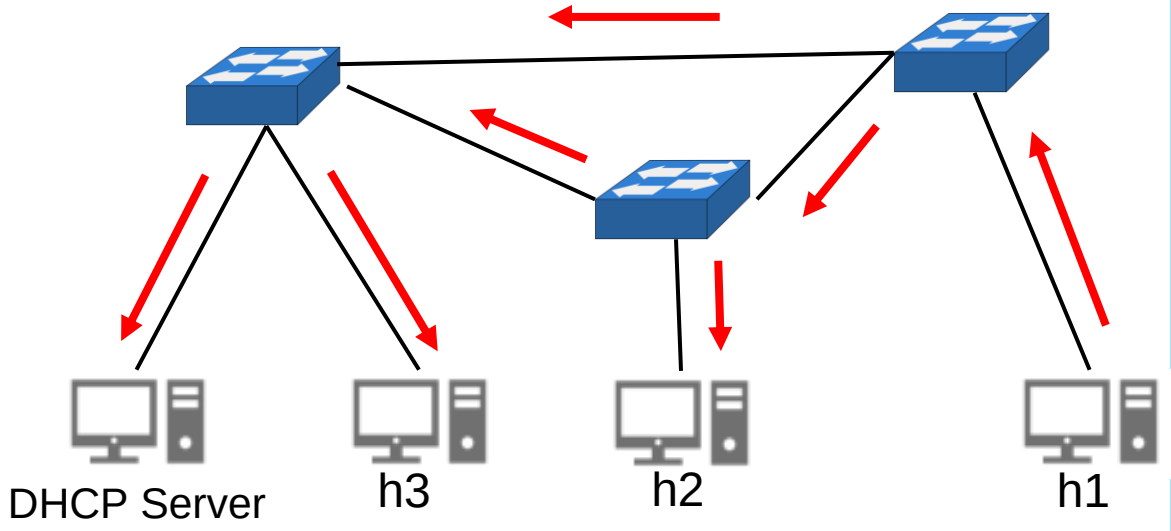
DHCP Workflow



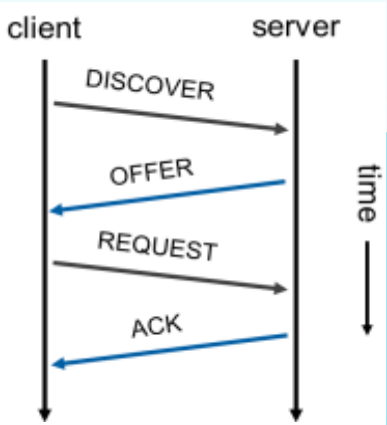
- When a host (e.g., h1) attaches to a network
  - Issues DHCPDISCOVER to locate available DHCP servers (broadcast)
- DHCP Servers receive DHCPDISCOVER
  - Reply DHCPOFFER (Broadcast in general, Unicast when renewing)
- Host (e.g., h1) chooses a server to reply DHCPREQUEST (broadcast)
- Server replies with DHCPACK (Broadcast in general, Unicast when renewing)

```
Src IP:  0.0.0.0
Dst IP:  255.255.255.255
Src MAC: <MAC of h1>
Dst MAC: ff:ff:ff:ff:ff:ff
```

DHCP DISCOVER



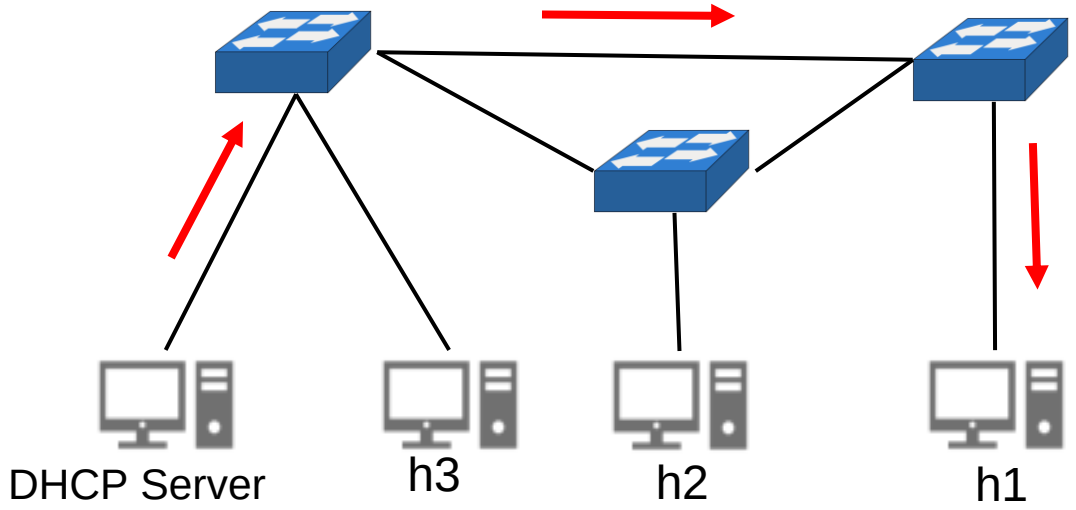
DHCP Workflow



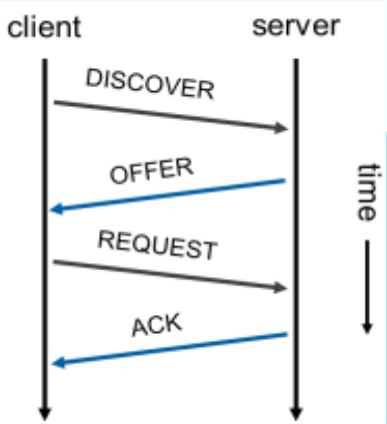
- When a host (e.g., h1) attaches to a network
  - Issues DHCPDISCOVER to locate available DHCP servers (broadcast)
- DHCP Servers receive DHCPDISCOVER
  - Reply DHCPOFFER (Broadcast in general, Unicast when renewing)
- Host (e.g., h1) chooses a server to reply DHCPREQUEST (broadcast)
- Server replies with DHCPACK (Broadcast in general, Unicast when renewing)

```
Src IP:  <IP of server>
Dst IP:  255.255.255.255
Src MAC: <MAC of server>
Dst MAC: <MAC of h1>
Your IP address: 10.0.0.2
Subnet Mask: 255.255.255.0
IP Address Lease Time: 3600
```

DHCP OFFER



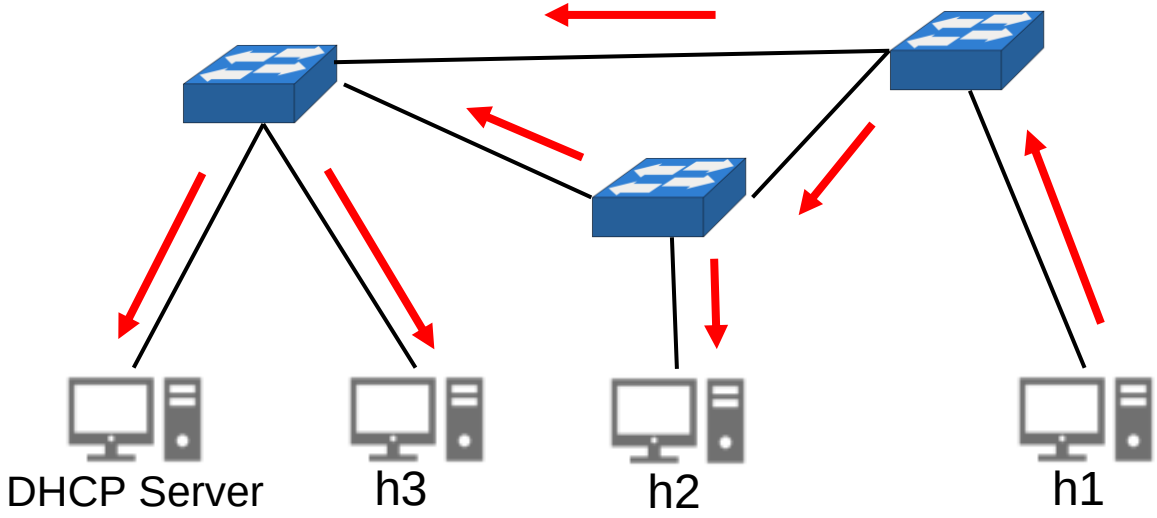
DHCP Workflow



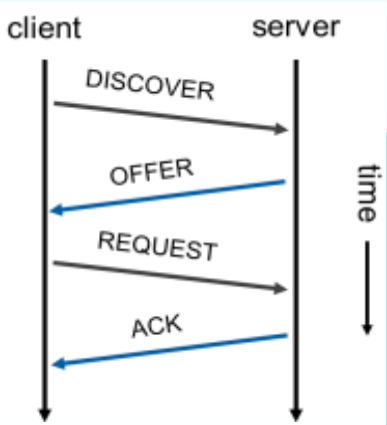
- When a host (e.g., h1) attaches to a network
  - Issues DHCPDISCOVER to locate available DHCP servers (broadcast)
- DHCP Servers receive DHCPDISCOVER
  - Reply DHCPOFFER (Broadcast in general, Unicast when renewing)
- Host (e.g., h1) chooses a server to reply DHCPREQUEST (broadcast)
- Server replies with DHCPACK (Broadcast in general, Unicast when renewing)

```
Src IP:  0.0.0.0
Dst IP:  255.255.255.255
Src MAC: <MAC of h1>
Dst MAC: ff:ff:ff:ff:ff:ff
Requested IP address: 10.0.0.2
DHCP Server Identifier: <server IP>
```

DHCP REQUEST



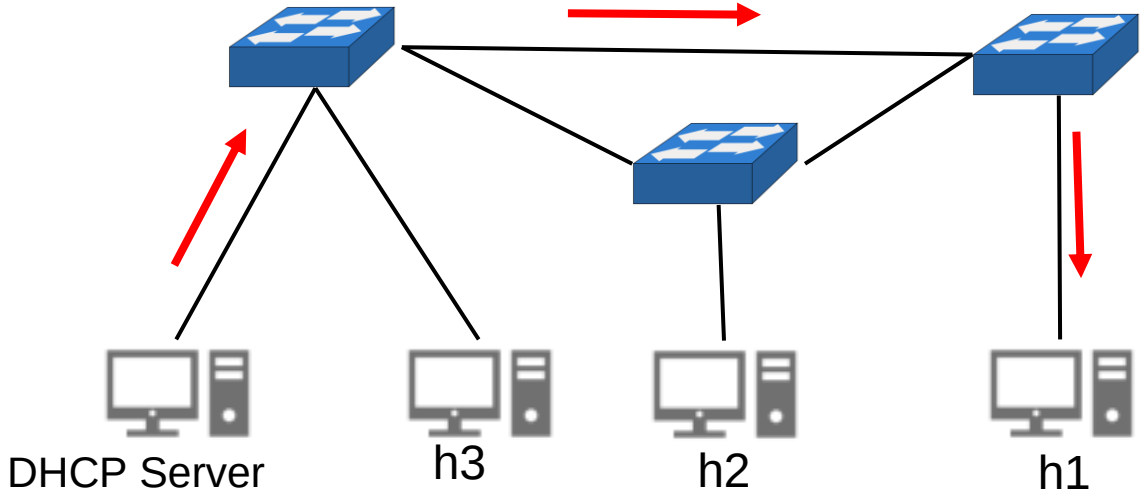
DHCP Workflow



- When a host (e.g., h1) attaches to a network
  - Issues DHCPDISCOBER to locate available DHCP servers (broadcast)
- DHCP Servers receive DHCPDISCOVER
  - Reply DHCPOFFER (Broadcast in general, Unicast when renewing)
- Host (e.g., h1) chooses a server to reply DHCPREQUEST (broadcast)
- Server replies with DHCPACK (Broadcast in general, Unicast when renewing)

```
Src IP:  <IP of server>
Dst IP:  255.255.255.255
Src MAC: <MAC of server>
Dst MAC: <MAC of h1>
Your IP address: 10.0.0.2
Subnet Mask: 255.255.255.0
IP Address Lease Time: 3600
```

DHCP ACK



Q & A