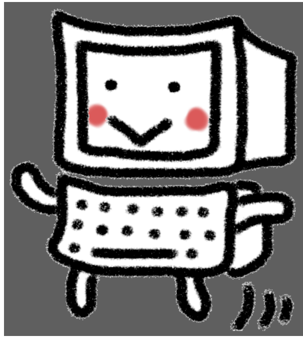
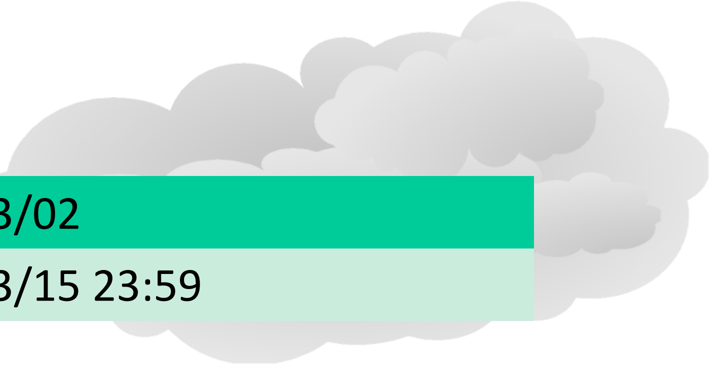


Lab 2

Packet Forwarding and DHCP



Data :	2021/03/02
Deadline :	2021/03/15 23:59

Outline

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

Objective

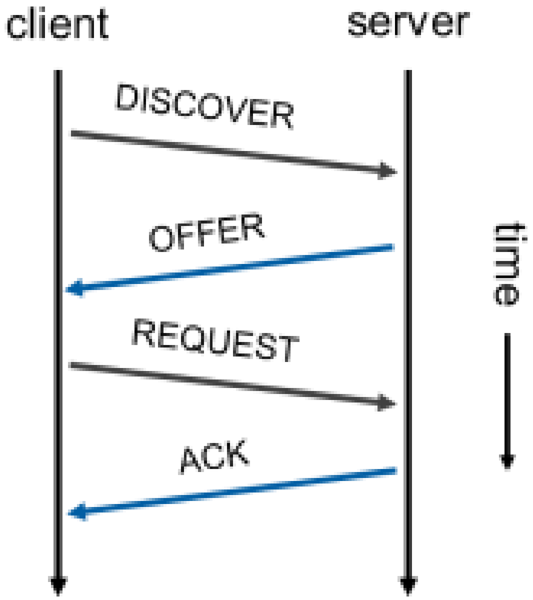
- Subnetting and Netmask
- Routing Rule Static Configuration
- DHCP Server configuration
- DHCP 4-Way Handshaking Messages
- Traceroute Observation

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 applications
- 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}/dhclient* lrw,' \
                       /etc/apparmor.d/sbin.dhclient
```

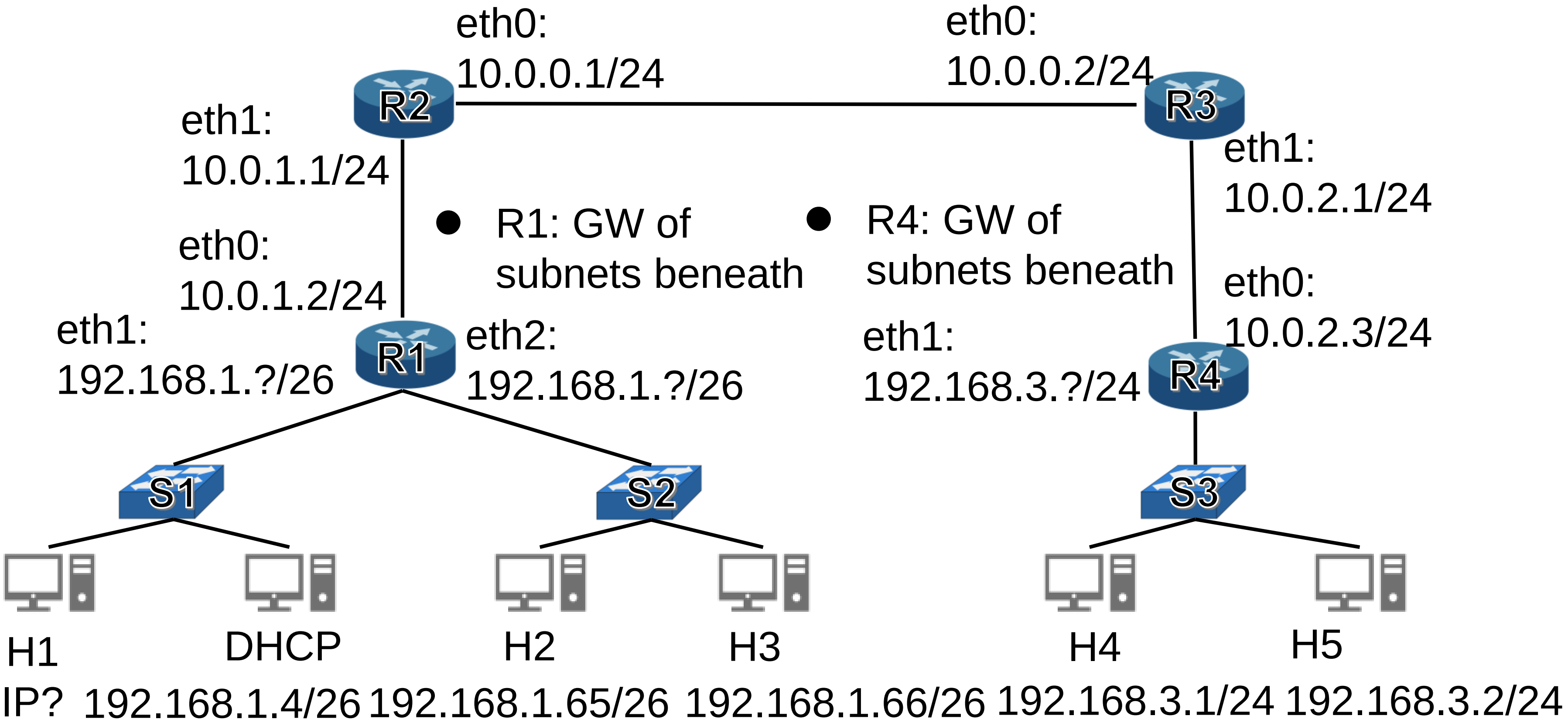
- Start AppArmor after setting

```
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



R2 eth0: 10.0.0.1/24

R3 eth0: 10.0.0.2/24

R2 eth1: 10.0.1.1/24

R3 eth1: 10.0.2.1/24

- R1: GW of subnets beneath
- R4: GW of subnets beneath

R1 eth0: 10.0.1.2/24

R4 eth0: 10.0.2.3/24

R1 eth1: 192.168.1.?/26

R1 eth2: 192.168.1.?/26

R4 eth1: 192.168.3.?/24

S1 S2 S3

H1 IP?

DHCP 192.168.1.4/26

H2 192.168.1.65/26

H3 192.168.1.66/26

H4 192.168.3.1/24

H5 192.168.3.2/24

Outline

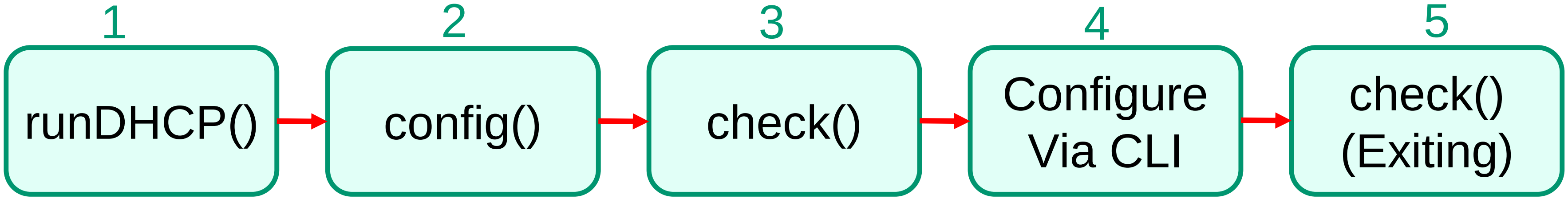
- Objective
- Introduction to DHCP
- Lab environment
  - Environment
  - DHCP utility setup
  - Lab topology
  - Python script for lab topology
- Lab requirement
- Appendix

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

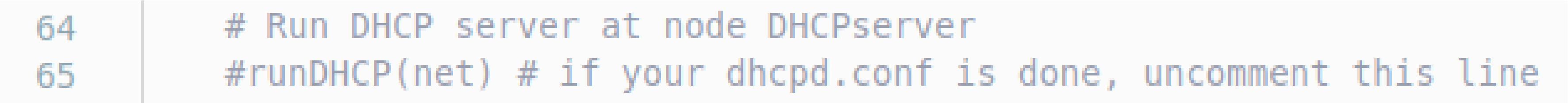
```
bash$ sudo python topology.py
```

- Components and Sequence of topology.py

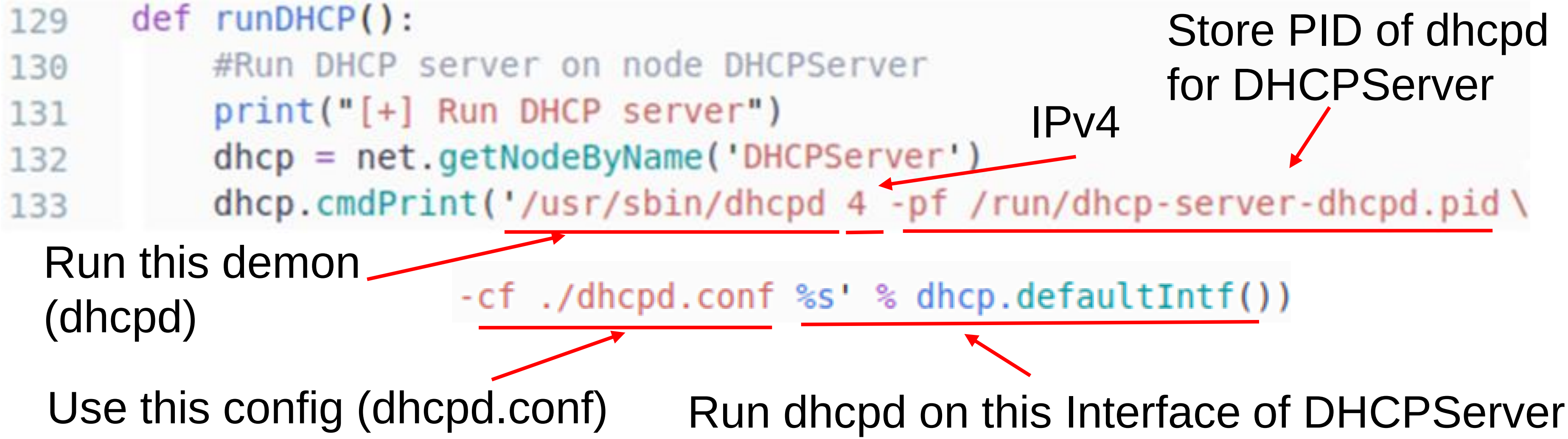


1. topology.py – runDHCP()

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



```
    # Run DHCP server at node DHCPserver
    #runDHCP(net) # if your dhcpd.conf is done, uncomment this line
```



```
def runDHCP():
    #Run DHCP server on node DHCPServer
    print("[+] Run DHCP server")
    dhcp = net.getNodeByName('DHCPServer')
    dhcp.cmdPrint('/usr/sbin/dhcpd 4 -pf /run/dhcp-server-dhcpd.pid \
            -cf ./dhcpd.conf %s' % dhcp.defaultIntf())
```

Store PID of dhcpd for DHCPServer

IPv4

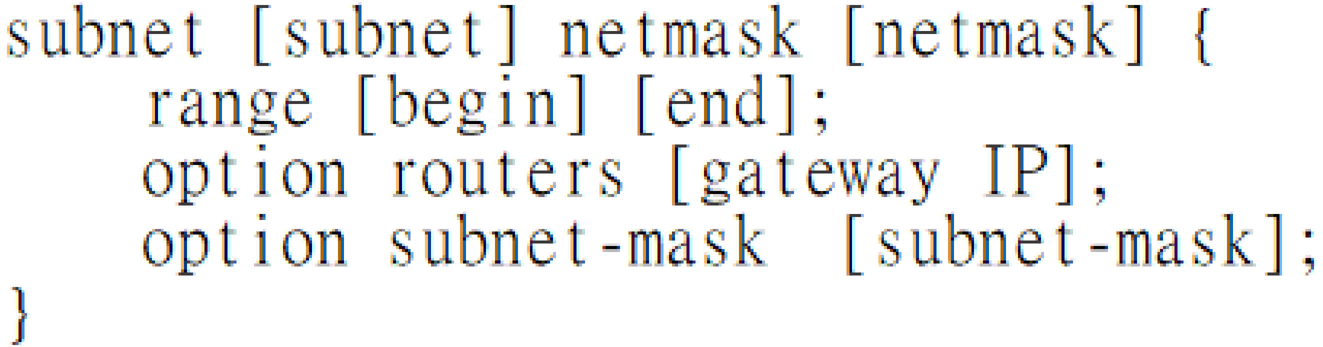
Run this demon (dhcpd)

Use this config (dhcpd.conf)

Run dhcpd on this Interface of DHCPServer

DHCP Configuration file – dhcpd.conf

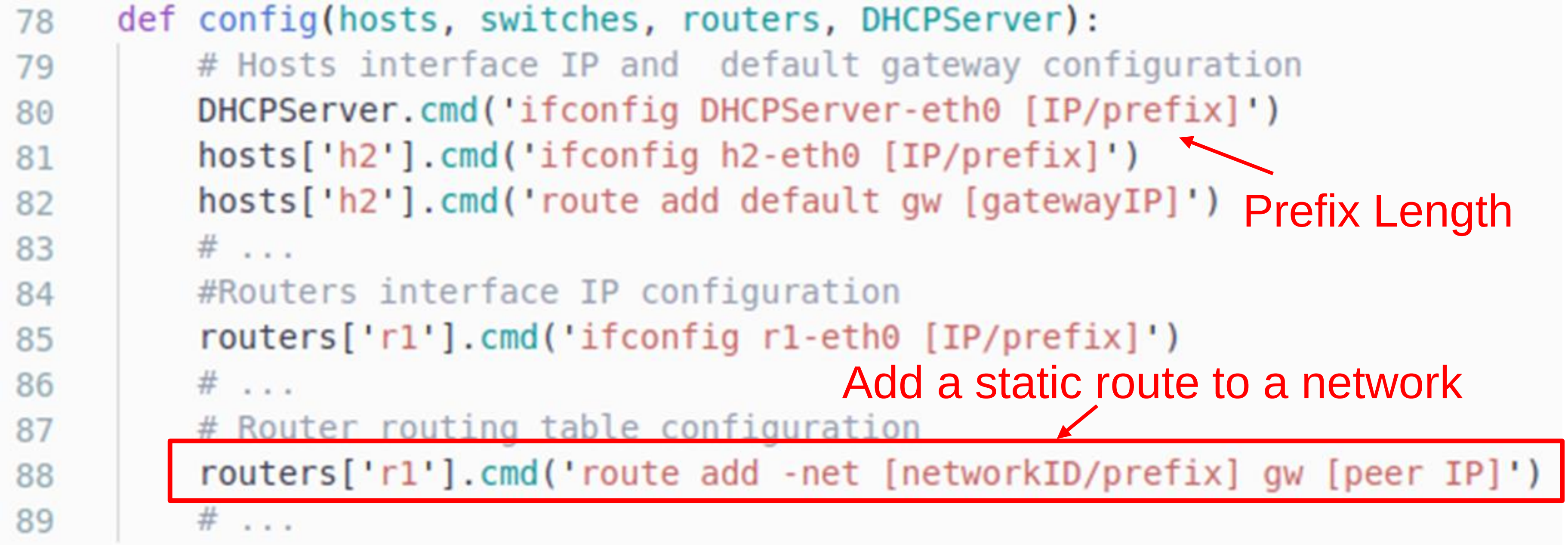
- Create dhcpd.conf in the same directory as topology.py
- dhcpd.conf



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

2. topology.py – config()

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



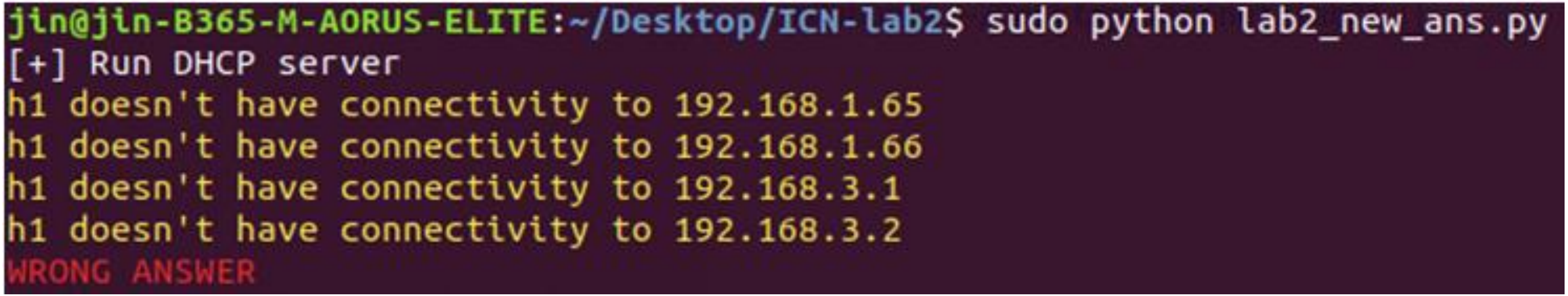
```
def config(hosts, switches, routers, DHCPServer):
    # Hosts interface IP and  default gateway configuration
    DHCPServer.cmd('ifconfig DHCPServer-eth0 [IP/prefix]')
    hosts['h2'].cmd('ifconfig h2-eth0 [IP/prefix]')
    hosts['h2'].cmd('route add default gw [gatewayIP]')
    # ...
    #Routers interface IP configuration
    routers['r1'].cmd('ifconfig r1-eth0 [IP/prefix]')
    # ...
    # Router routing table configuration
    routers['r1'].cmd('route add -net [networkID/prefix] gw [peer IP]')
    # ...
```

Prefix Length

Add a static route to a network

3. topology.py – check()

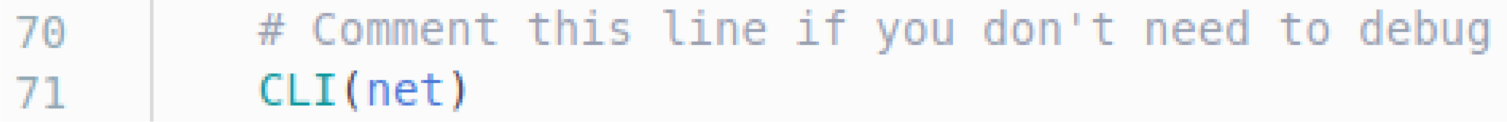
- check()
  - Script that check the correctness of your configuration until now
  - Recall: h1 does not have IP yet
  - All hosts except h1 should be able to ping one another
  - Check starts from h1 to other hosts, then the next to the remaining hosts
  - Print WORNG ANSWER if fails



```
jin@jin-B365-M-AORUS-ELITE:~/Desktop/ICN-lab2$ sudo python lab2_new_ans.py
[+] Run DHCP server
h1 doesn't have connectivity to 192.168.1.65
h1 doesn't have connectivity to 192.168.1.66
h1 doesn't have connectivity to 192.168.3.1
h1 doesn't have connectivity to 192.168.3.2
WRONG ANSWER
```

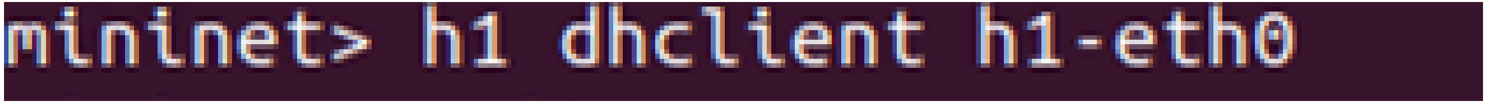
4. topology.py – Configure by CLI

- Launch mininet CLI



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

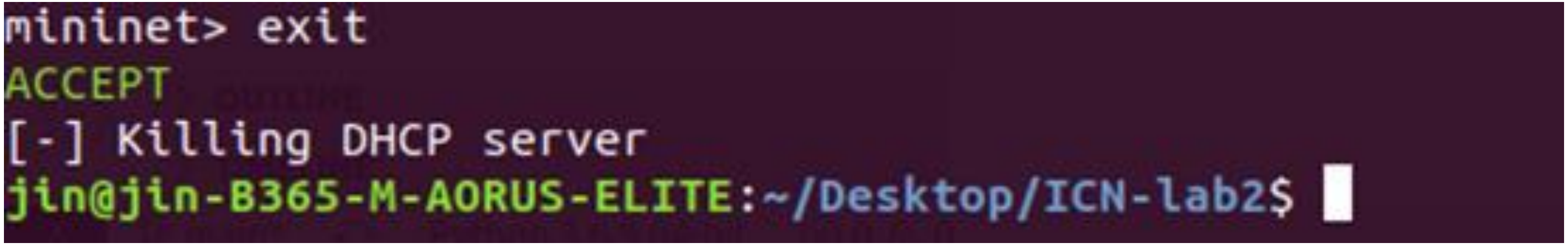
  - to Debug
    - ping hosts, traceroute, …
  - To perform more configuration
    - add routing rules, change IPs, …
- E.g., Configure IP and Gateway of h1
  - Run DHCP Client on h1 with eth0



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

5. topology.py – check() (Exiting)

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



```
mininet> exit
ACCEPT
[-] Killing DHCP server
jin@jin-B365-M-AORUS-ELITE:~/Desktop/ICN-lab2$ 
```

  - All hosts should now reach one another
    - Print ACCEPT

Outline

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

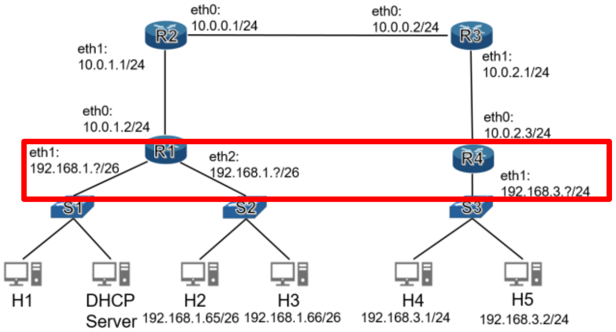
Step 1: Complete topology.py

- 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 h2 ping h3? Briefly explain why or why not. (5%)
    b) Can h2 ping h4? Briefly explain why or why not. (5%)

Complete topology.py so that all hosts, except h1, 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 192.168.1.0/26
    - Default gateway for 192.168.1.0/26
- Run dhcp on h1

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

Part 2 Questions

3. Capture DHCP messages and show the IPs and MACs (10%)

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

4. Can hosts other than h1 acquire IP addresses from DHCP server? Briefly explain your answer. (10%)

Part 3 Questions

- 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
```

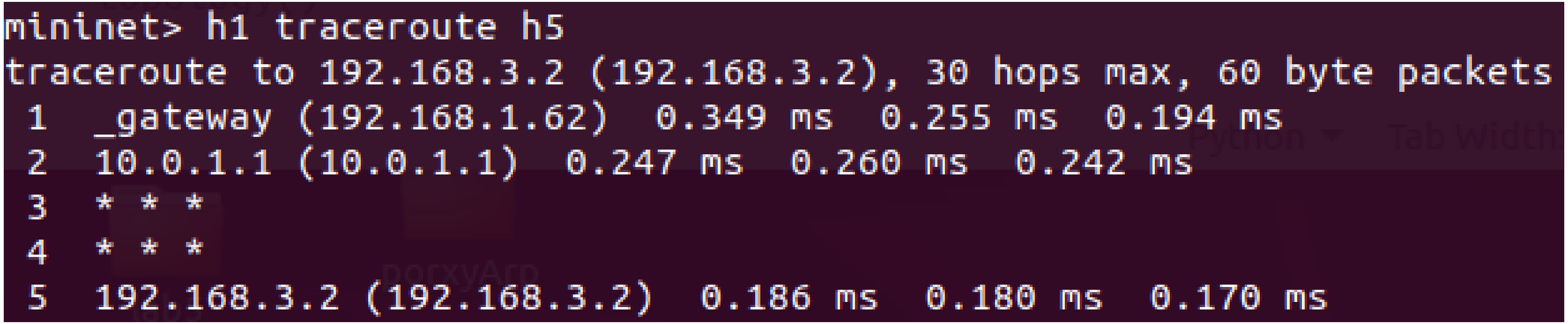
e. What does r1 do on the packets from h1 to h5, and h5 to h1, respectively? Capture packets to explain your answers.

Part 3 Questions

- Activate Wireshark on h1-eth0, and execute traceroute on h1

```
mininet> h1 wireshark &   #listen at h1-eth0
mininet> h1 traceroute h5
```

- Ultimately, traceroute on h1 may show the following hop details



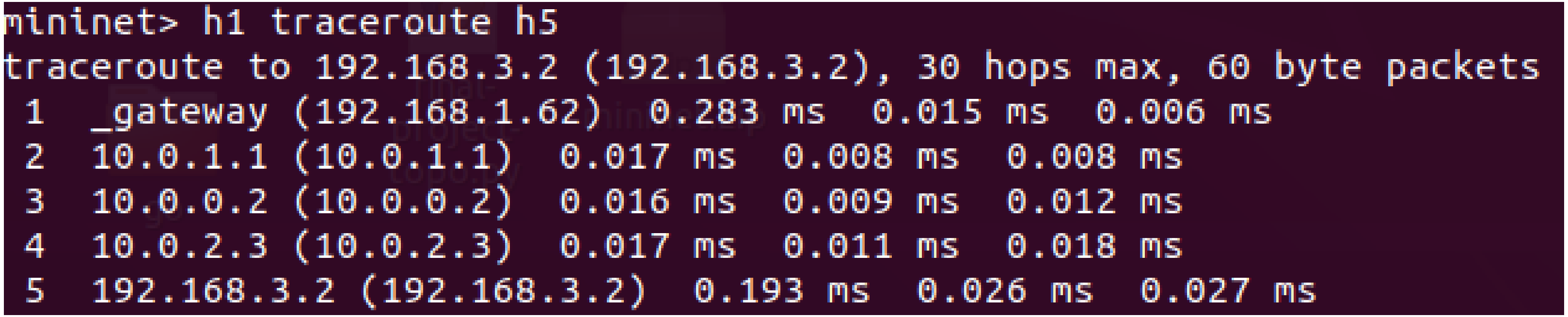
```
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.62)  0.349 ms  0.255 ms  0.194 ms
 2  10.0.1.1 (10.0.1.1)  0.247 ms  0.260 ms  0.242 ms
 3  * * *
 4  * * *
 5  192.168.3.2 (192.168.3.2)  0.186 ms  0.180 ms  0.170 ms
```

Part 3 Questions (cont.)

7. Capture all ICMP messages received by h1 and explain why h1 can only derive only 1st, 2nd, and 5th hops details. (10%)
8. H1 uses some ICMP messages to derive 1st and 2nd hop details. What are the type(s) and sender(s) of the ICMP messages? (5%)
9. H1 uses some ICMP messages to derive 5th hop details. What are the type(s) and sender(s) of the ICMP messages? (5%)

Bonus (10%)

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

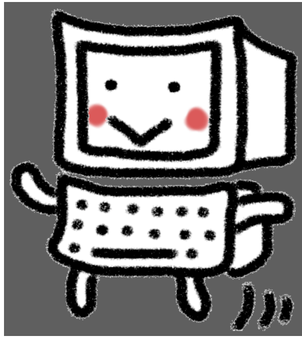


```
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.62)  0.283 ms  0.015 ms  0.006 ms
 2  10.0.1.1 (10.0.1.1)  0.017 ms  0.008 ms  0.008 ms
 3  10.0.0.2 (10.0.0.2)  0.016 ms  0.009 ms  0.012 ms
 4  10.0.2.3 (10.0.2.3)  0.017 ms 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

- Try to configure the nodes so that traceroute can output the above hop details.

Report Submission

- Files
  - <studentID>_topology.py (20%)
  - dhcpd.conf (20%)
  - A report: lab2_<studentID>.pdf (60%+10% Bonus)
    - Question Answers
- Submit
  - Zip files into a zip file
    - Name: lab2_<studentID>.zip



Q & A

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

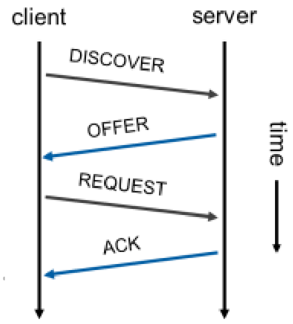
  ```
  bash$ ifconfig
  ```

Appendix

- dhcpd.conf man page
  - https://linux.die.net/man/5/dhcpd.conf

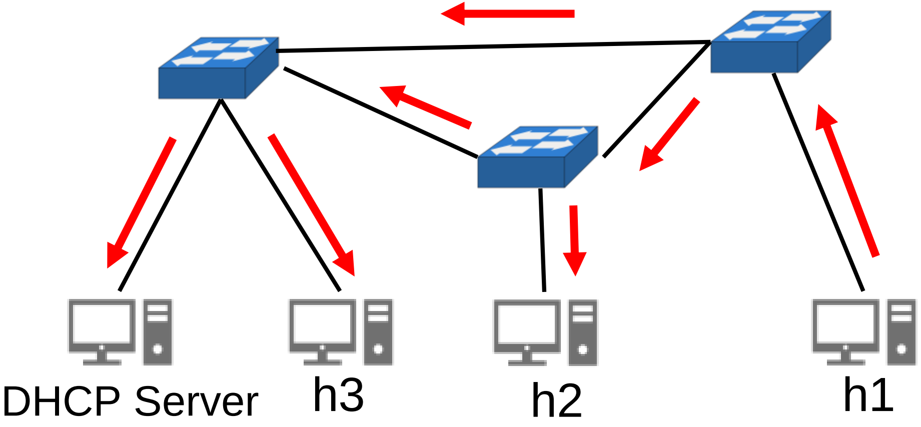
DHCP Workflow

- 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 (broadcas
- Server replies with DHCPACK (Broadcast in general, Unicast when renewing)



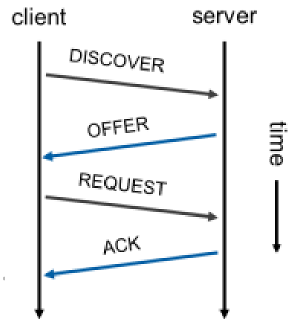
```
Src IP:   0.0.0.0
Dst IP:   255.255.255.255
Src MAC: <MAC of h1>
Dst MAC: ff:ff:ff:ff:ff:ff
```

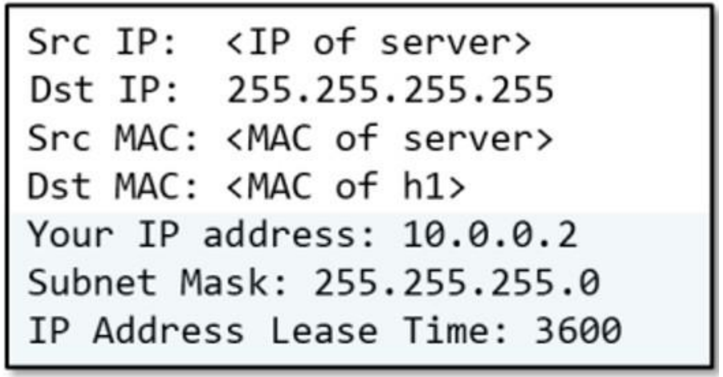
DHCP DISCOVER



DHCP Workflow

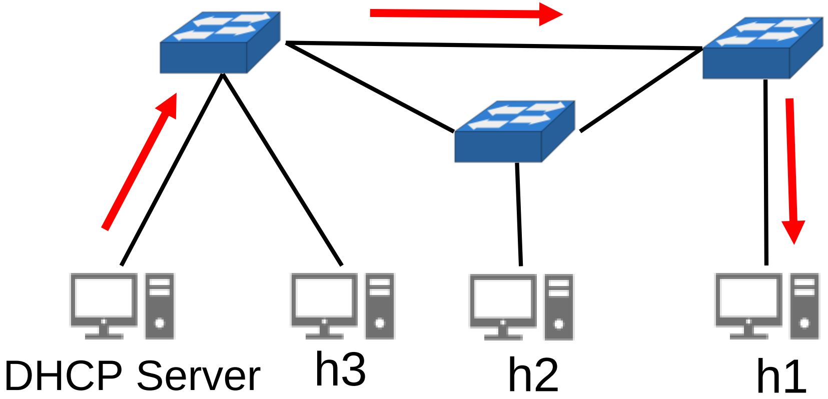
- When a host (e.g., h1) attaches to a network
  - Issues DHCPDISCOBER to locate available DHCP servers (broadcast)
- DHCP Servers receive DHCPDISCOVER
  - Reply DHCPOFFER (Broadcast in general, Unicast when renewing)
- Host (e.g., h1) chooses a server to reply DHCPREQUEST (broadcas
- Server replies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





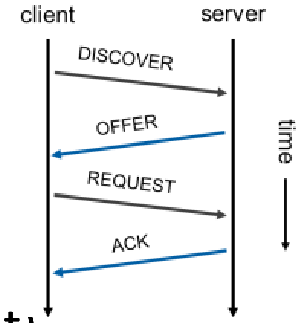
```
Src IP:   <IP of server>
Dst IP:   255.255.255.255
Src MAC: <MAC of server>
Dst MAC: <MAC of h1>
Your IP address: 10.0.0.2
Subnet Mask: 255.255.255.0
IP Address Lease Time: 3600
```

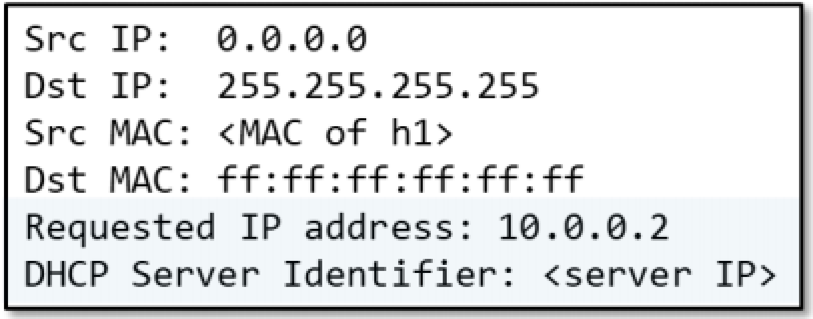
DHCP OFFER



DHCP Workflow

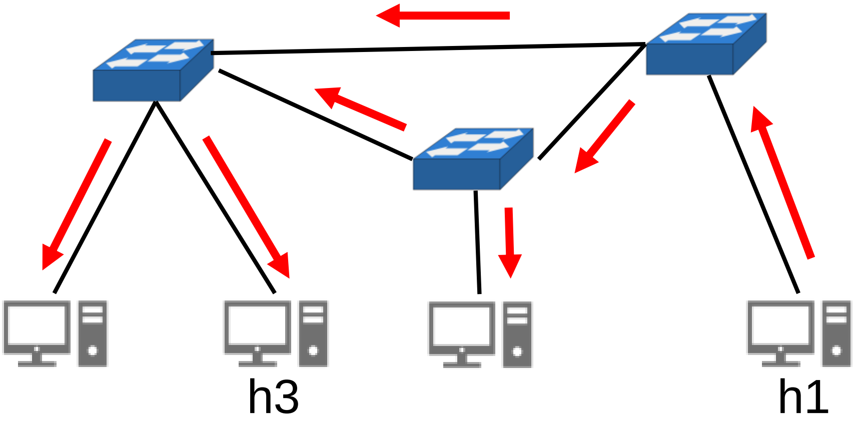
- When a host (e.g., h1) attaches to a network
  - Issues DHCPDISCOBER to locate available DHCP servers (broadcast)
- DHCP Servers receive DHCPDISCOVER
  - Reply DHCPOFFER (Broadcast in general, Unicast when renewing)
- Host (e.g., h1) chooses a server to reply DHCPREQUEST (broadcast)
- Server replies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



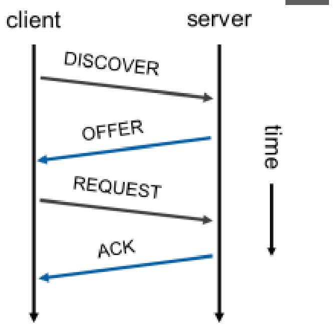


```
Src IP:  0.0.0.0
Dst IP:  255.255.255.255
Src MAC: <MAC of h1>
Dst MAC: ff:ff:ff:ff:ff:ff
Requested IP address: 10.0.0.2
DHCP Server Identifier: <server IP>
```

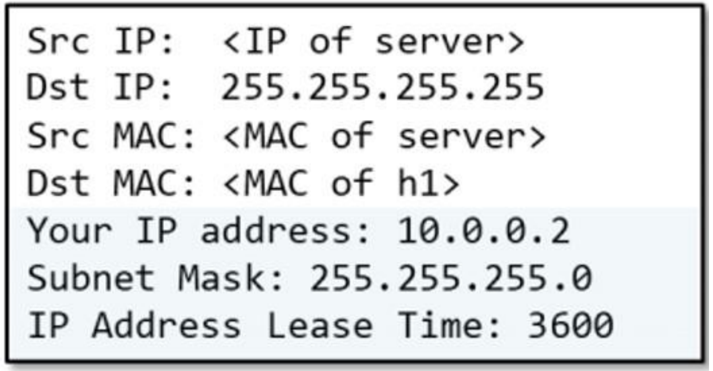
DHCP REQUEST



DHCP Workflow

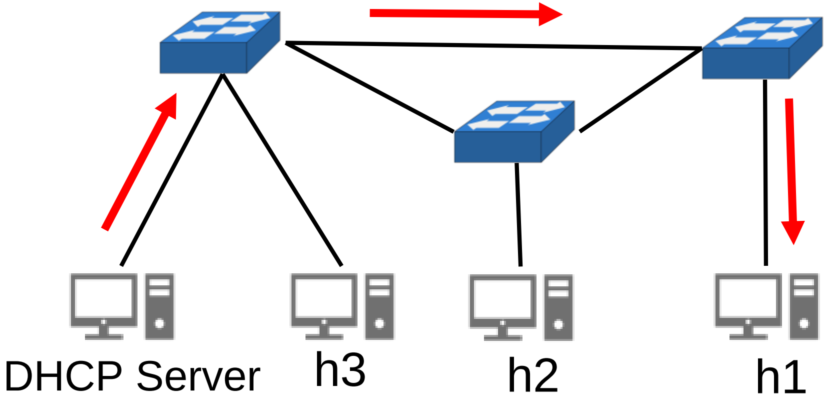


- When a host (e.g., h1) attaches to a network
  - Issues DHCPDISCOBER to locate available DHCP servers (broadcast)
- DHCP Servers receive DHCPDISCOVER
  - Reply DHCPOFFER (Broadcast in general, Unicast when renewing)
- Host (e.g., h1) chooses a server to reply DHCPREQUEST (broadcast)
- Server replies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


DHCP Server
h3
h2
h1