ARP, Wireshark, Netsim

Cloud networking

1.ARP (linux.cs.pdx.edu)

- Include both in your lab notebook

```
raghuram@ada:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:13:a0:c6 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 131.252.208.103/24 brd 131.252.208.255 scope global dynamic ens3
       valid_lft 10279sec preferred_lft 10279sec
```

- What is the default router's IP address (e.g. the gateway address for the default route 0.0.0.0/0)

```
raghuram@ada:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         131.252.208.1   0.0.0.0         UG        0 0          0 ens3
131.252.208.0   0.0.0.0         255.255.255.0   U         0 0          0 ens3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 ens3
```

- What is the name of the default router and its hardware address?

Name: router.seas.pdx.edu **Hardware address:** 00:00:5e:00:01:01

- How many entries are there in the ARP table?

```
raghuram@ada:~$ arp -a | wc -l
42
```

2. -

- List any IP addresses share the same hardware address

? (131.252.208.250) at e0:89:9d:a8:0a:dd [ether] on ens3
? (169.254.169.254) at e0:89:9d:a8:0a:dd [ether] on ens3

cs299lab.cs.pdx.edu (131.252.208.86) at 52:54:00:a3:46:7f [ether] on ens3
quizor6.cs.pdx.edu (131.252.208.60) at 52:54:00:a3:46:7f [ether] on ens3

- How many less hardware addresses are there than IP addresses in the ARP table?

  42-40 =2

- Include the command in your lab notebook

  arp -an | awk -F '[()]' '{print $2}' > arp_entries

- What network prefix do most of the IP addresses in the ARP table share?

```
raghuram@ada:~$ cat arp_entries
131.252.208.121
131.252.208.13
131.252.208.36
131.252.208.60
131.252.208.84
131.252.208.38
```

3. ARP (Cloud)

- Include both in your lab notebook

```
raghuram@course-vm:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group default qlen 1000
    link/ether 42:01:0a:8a:00:02 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 10.138.0.2/32 metric 100 scope global dynamic ens4
       valid_lft 86362sec preferred_lft 86362sec
    inet6 fe80::4001:aff:fe8a:2/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:1f:89:ae:4a brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

- What is the default router's IP address (e.g. the gateway address for the default route 0.0.0.0/0)

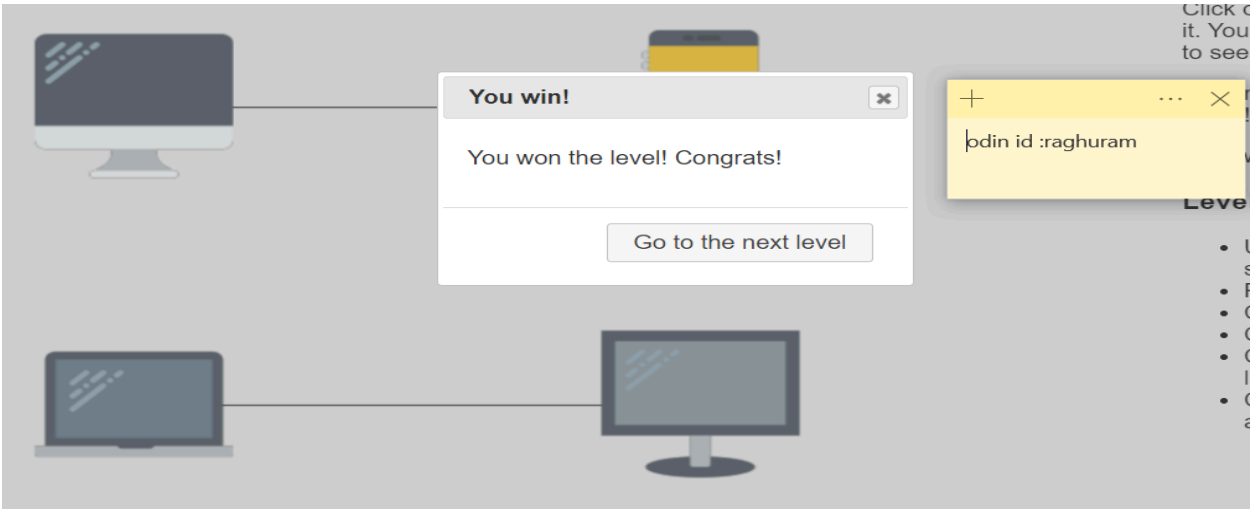
```
raghuram@course-vm:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.138.0.1      0.0.0.0         UG        0 0          0 ens4
10.138.0.1      0.0.0.0         255.255.255.255 UH        0 0          0 ens4
169.254.169.254 10.138.0.1      255.255.255.255 UGH       0 0          0 ens4
172.17.0.0      0.0.0.0         255.255.0.0     U         0 0          0 docker0
```

- What is the default router's hardware address?

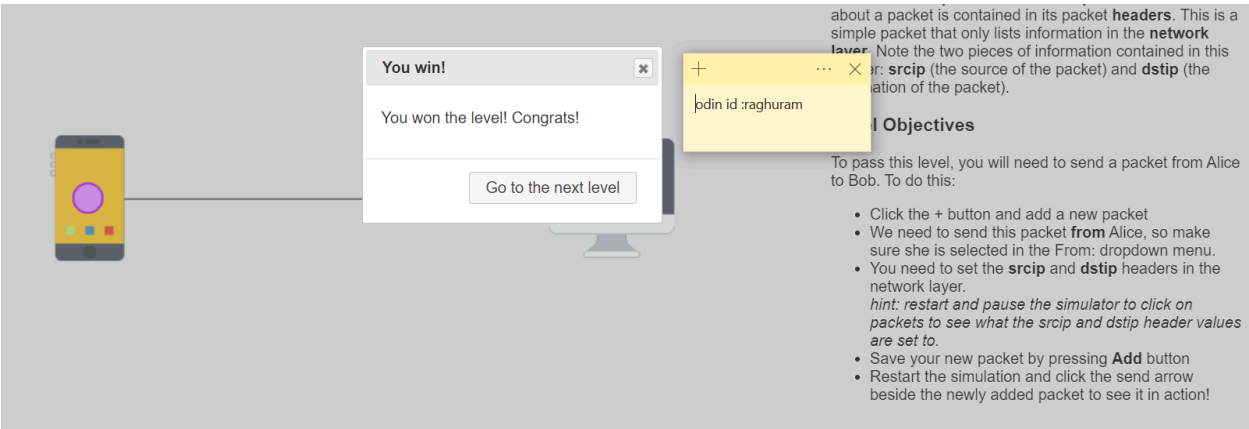
```
raghuram@course-vm:~$ arp 10.138.0.1
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   42:01:0a:8a:00:01   C                     ens4
```

4. Netsim

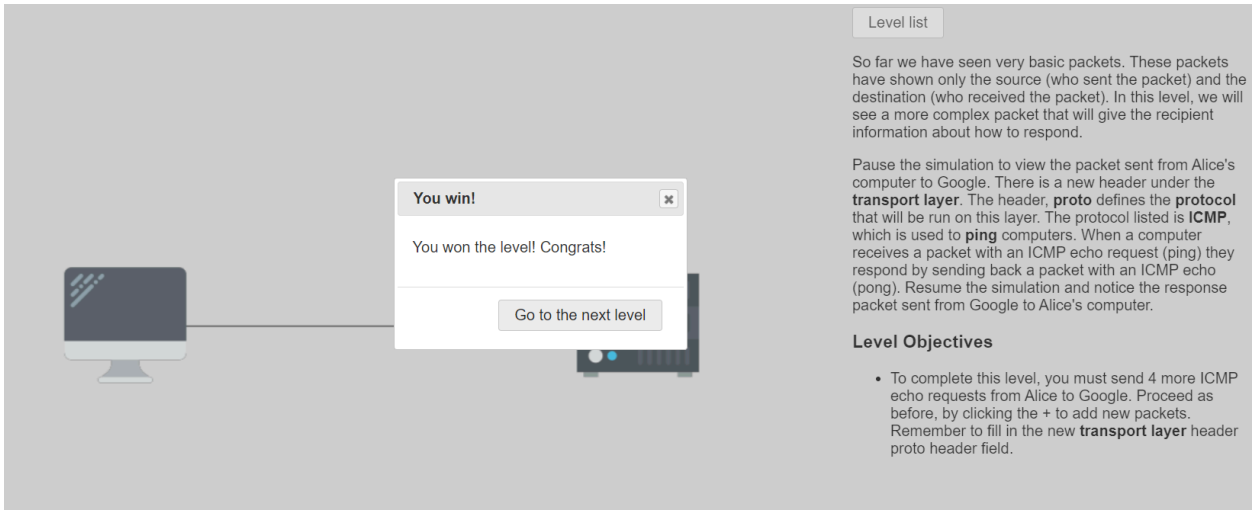
1


You win!
You won the level! Congrats!
Go to the next level
odin id :raghuram

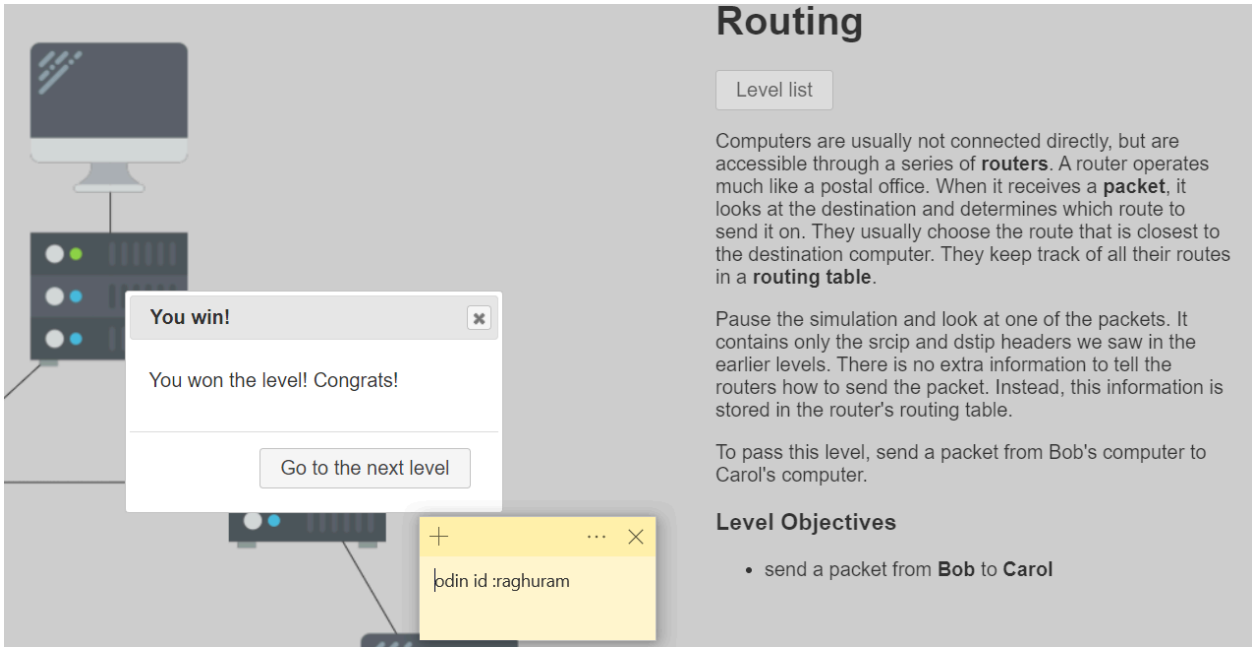
2


You win!
You won the level! Congrats!
Go to the next level
odin id :raghuram
about a packet is contained in its packet headers. This is a simple packet that only lists information in the network
srcip (the source of the packet) and dstip (the
ation of the packet).
Objectives
To pass this level, you will need to send a packet from Alice to Bob. To do this:
Click the + button and add a new packet
We need to send this packet from Alice, so make sure she is selected in the From: dropdown menu.
You need to set the srcip and dstip headers in the network layer.
hint: restart and pause the simulator to click on packets to see what the srcip and dstip header values are set to.
Save your new packet by pressing Add button
Restart the simulation and click the send arrow beside the newly added packet to see it in action!

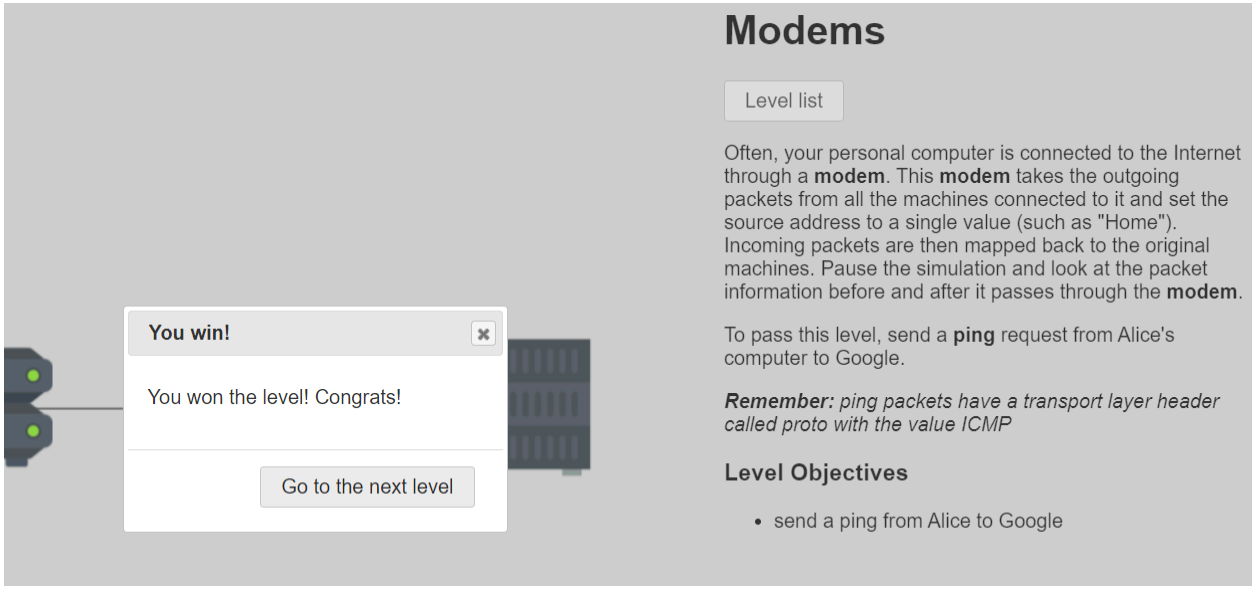
3


You win!
You won the level! Congrats!
Go to the next level
Level list
So far we have seen very basic packets. These packets have shown only the source (who sent the packet) and the destination (who received the packet). In this level, we will see a more complex packet that will give the recipient information about how to respond.
Pause the simulation to view the packet sent from Alice's computer to Google. There is a new header under the transport layer. The header, proto defines the protocol that will be run on this layer. The protocol listed is ICMP, which is used to ping computers. When a computer receives a packet with an ICMP echo request (ping) they respond by sending back a packet with an ICMP echo (pong). Resume the simulation and notice the response packet sent from Google to Alice's computer.
Level Objectives
To complete this level, you must send 4 more ICMP echo requests from Alice to Google. Proceed as before, by clicking the + to add new packets. Remember to fill in the new transport layer header proto header field.

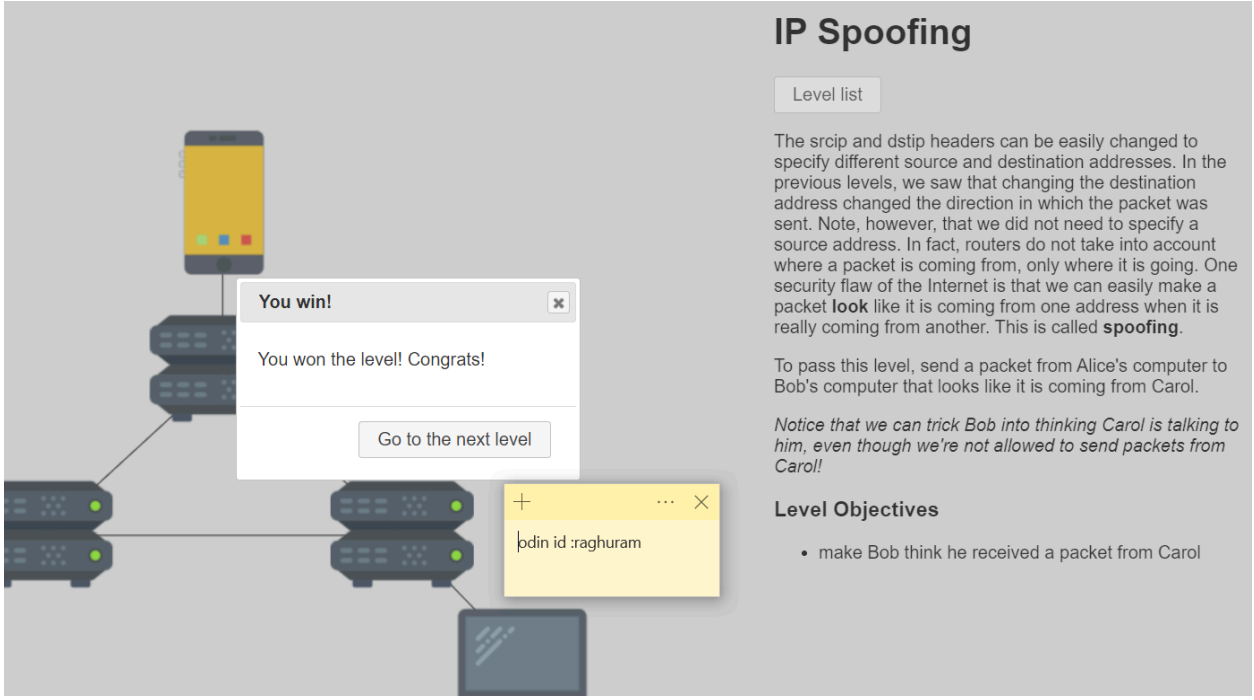
4


You win!
You won the level! Congrats!
Go to the next level
odin id :raghuram
Routing
Level list
Computers are usually not connected directly, but are accessible through a series of routers. A router operates much like a postal office. When it receives a packet, it looks at the destination and determines which route to send it on. They usually choose the route that is closest to the destination computer. They keep track of all their routes in a routing table.
Pause the simulation and look at one of the packets. It contains only the srcip and dstip headers we saw in the earlier levels. There is no extra information to tell the routers how to send the packet. Instead, this information is stored in the router's routing table.
To pass this level, send a packet from Bob's computer to Carol's computer.
Level Objectives
send a packet from Bob to Carol

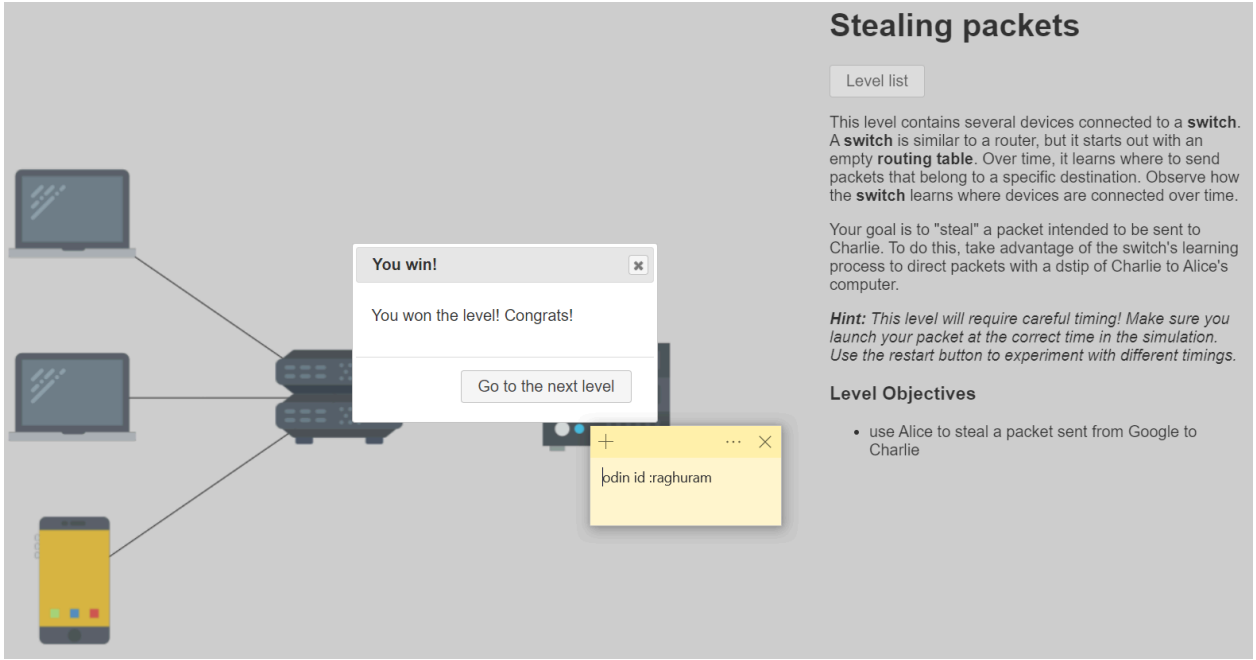
5


You win!
You won the level! Congrats!
Go to the next level
Modems
Level list
Often, your personal computer is connected to the Internet through a **modem**. This **modem** takes the outgoing packets from all the machines connected to it and set the source address to a single value (such as "Home"). Incoming packets are then mapped back to the original machines. Pause the simulation and look at the packet information before and after it passes through the **modem**.
To pass this level, send a **ping** request from Alice's computer to Google.
Remember: *ping packets have a transport layer header called proto with the value ICMP*
Level Objectives
• send a ping from Alice to Google

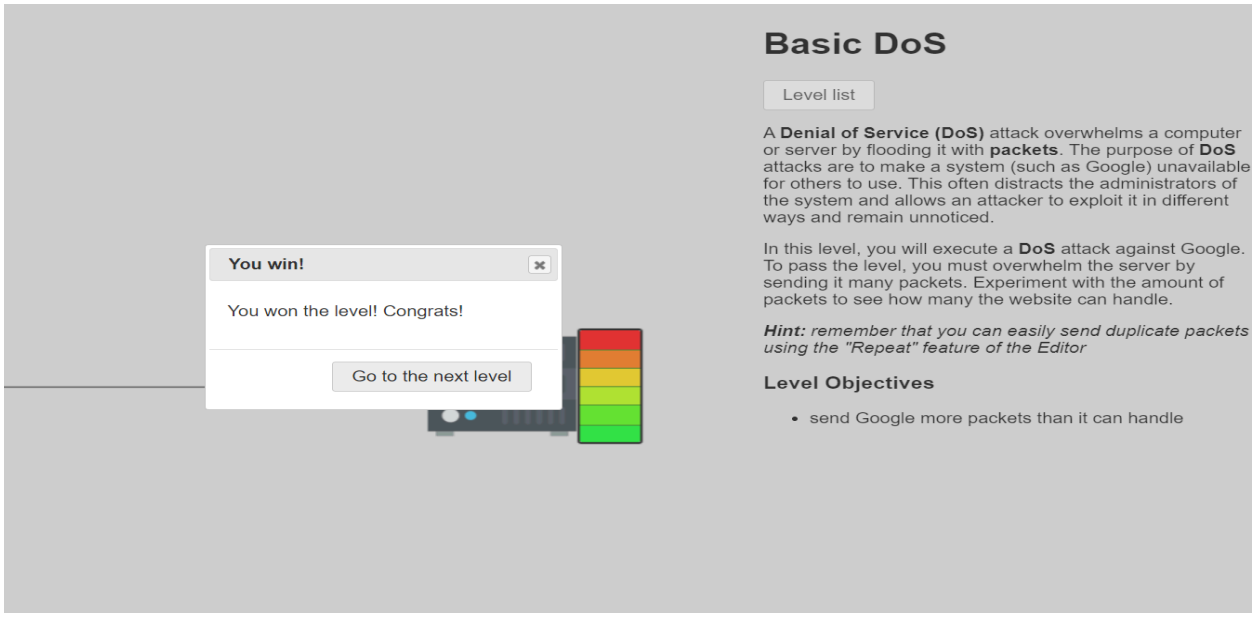
6


You win!
You won the level! Congrats!
Go to the next level
odin id :raghuram
IP Spoofing
Level list
The srcip and dstip headers can be easily changed to specify different source and destination addresses. In the previous levels, we saw that changing the destination address changed the direction in which the packet was sent. Note, however, that we did not need to specify a source address. In fact, routers do not take into account where a packet is coming from, only where it is going. One security flaw of the Internet is that we can easily make a packet **look** like it is coming from one address when it is really coming from another. This is called **spoofing**.
To pass this level, send a packet from Alice's computer to Bob's computer that looks like it is coming from Carol.
Notice that we can trick Bob into thinking Carol is talking to him, even though we're not allowed to send packets from Carol!
Level Objectives
• make Bob think he received a packet from Carol

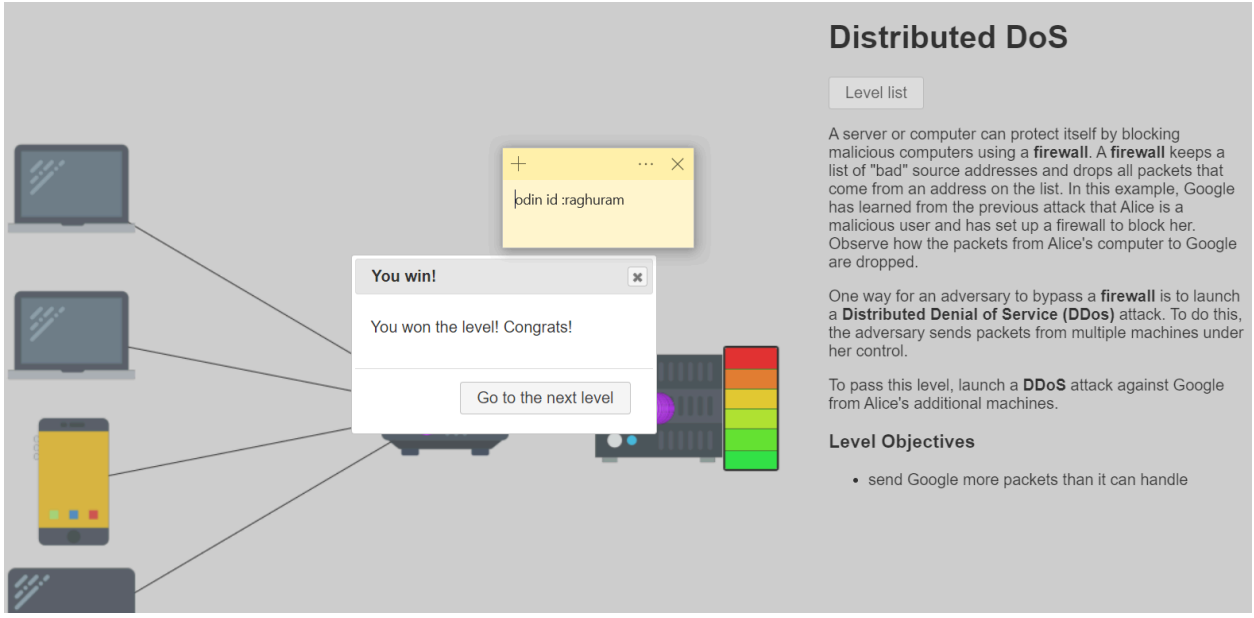
7


Stealing packets
Level list
This level contains several devices connected to a **switch**. A **switch** is similar to a router, but it starts out with an empty **routing table**. Over time, it learns where to send packets that belong to a specific destination. Observe how the **switch** learns where devices are connected over time.
Your goal is to "steal" a packet intended to be sent to Charlie. To do this, take advantage of the switch's learning process to direct packets with a dstip of Charlie to Alice's computer.
Hint: *This level will require careful timing! Make sure you launch your packet at the correct time in the simulation. Use the restart button to experiment with different timings.*
Level Objectives
• use Alice to steal a packet sent from Google to Charlie
You win!
You won the level! Congrats!
Go to the next level
odin id :raghuram

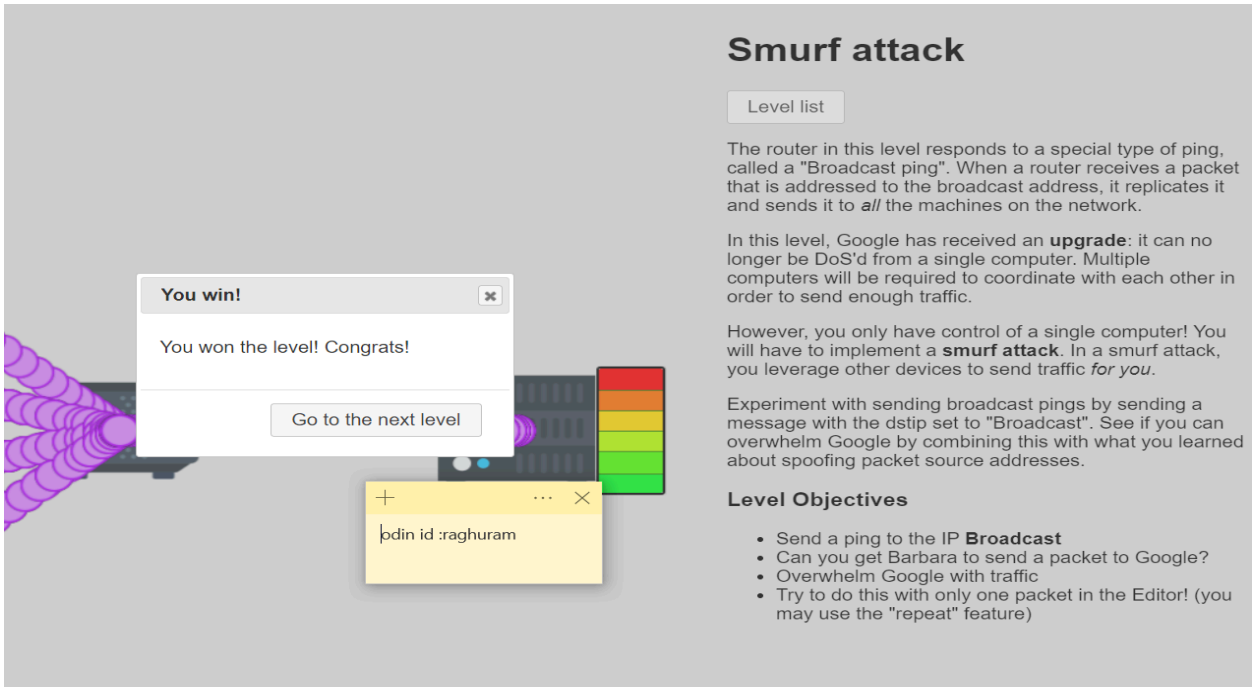
8


Basic DoS
Level list
A **Denial of Service (DoS)** attack overwhelms a computer or server by flooding it with **packets**. The purpose of **DoS** attacks are to make a system (such as Google) unavailable for others to use. This often distracts the administrators of the system and allows an attacker to exploit it in different ways and remain unnoticed.
In this level, you will execute a **DoS** attack against Google. To pass the level, you must overwhelm the server by sending it many packets. Experiment with the amount of packets to see how many the website can handle.
Hint: *remember that you can easily send duplicate packets using the "Repeat" feature of the Editor*
Level Objectives
• send Google more packets than it can handle
You win!
You won the level! Congrats!
Go to the next level

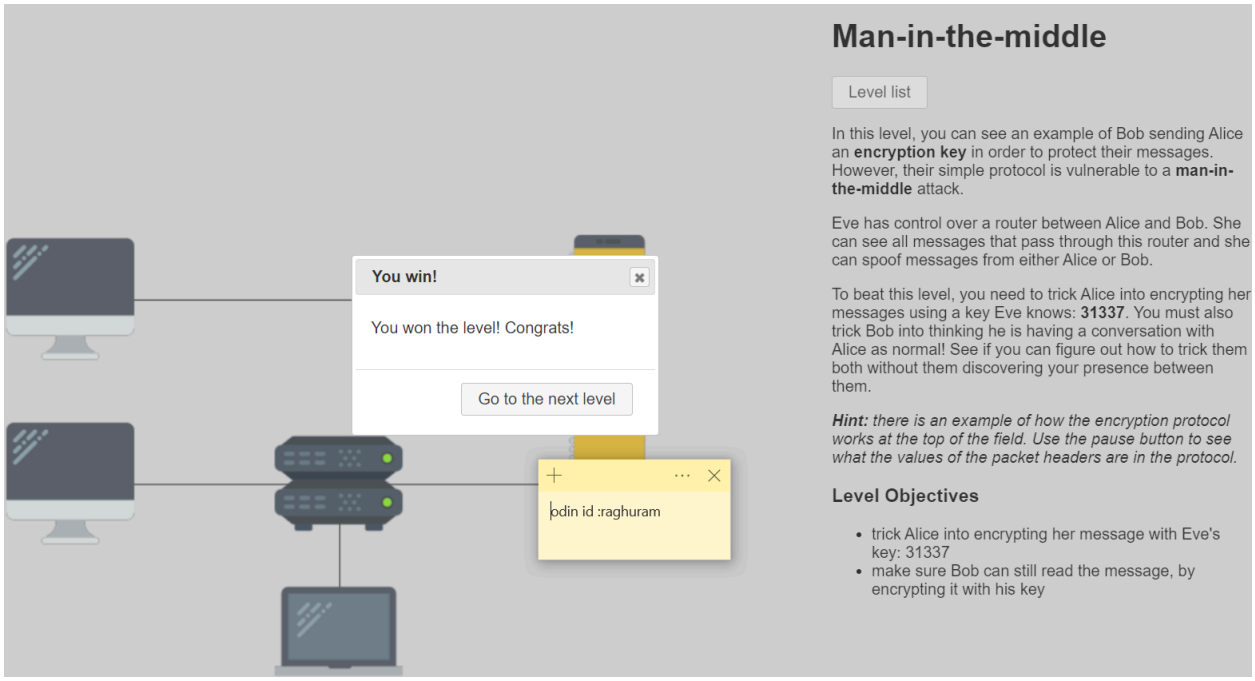
9


Distributed DoS
Level list
A server or computer can protect itself by blocking malicious computers using a **firewall**. A **firewall** keeps a list of "bad" source addresses and drops all packets that come from an address on the list. In this example, Google has learned from the previous attack that Alice is a malicious user and has set up a firewall to block her. Observe how the packets from Alice's computer to Google are dropped.
One way for an adversary to bypass a **firewall** is to launch a **Distributed Denial of Service (DDos)** attack. To do this, the adversary sends packets from multiple machines under her control.
To pass this level, launch a **DDoS** attack against Google from Alice's additional machines.
Level Objectives
send Google more packets than it can handle
odin id :raghuram
You win!
You won the level! Congrats!
Go to the next level

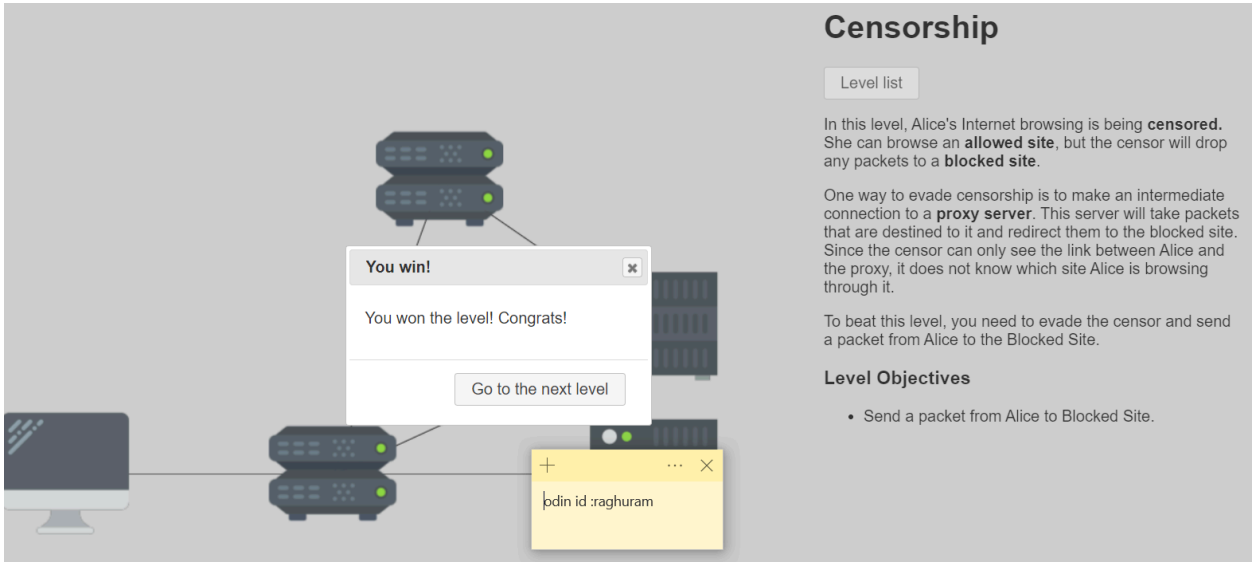
10


Smurf attack
Level list
The router in this level responds to a special type of ping, called a "Broadcast ping". When a router receives a packet that is addressed to the broadcast address, it replicates it and sends it to *all* the machines on the network.
In this level, Google has received an **upgrade**: it can no longer be DoS'd from a single computer. Multiple computers will be required to coordinate with each other in order to send enough traffic.
However, you only have control of a single computer! You will have to implement a **smurf attack**. In a smurf attack, you leverage other devices to send traffic *for you*.
Experiment with sending broadcast pings by sending a message with the dstip set to "Broadcast". See if you can overwhelm Google by combining this with what you learned about spoofing packet source addresses.
Level Objectives
Send a ping to the IP **Broadcast**
Can you get Barbara to send a packet to Google?
Overwhelm Google with traffic
Try to do this with only one packet in the Editor! (you may use the "repeat" feature)
You win!
You won the level! Congrats!
Go to the next level
odin id :raghuram

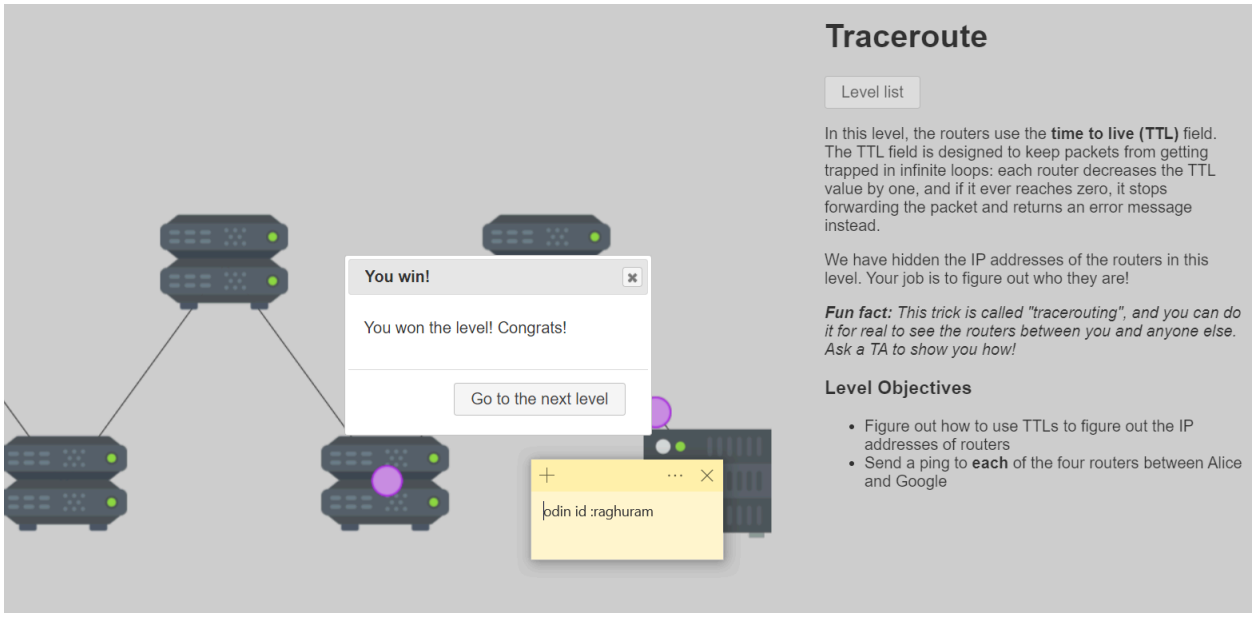
11


You win!
You won the level! Congrats!
Go to the next level
odin id :raghuram
Man-in-the-middle
Level list
In this level, you can see an example of Bob sending Alice an **encryption key** in order to protect their messages. However, their simple protocol is vulnerable to a **man-in-the-middle** attack.
Eve has control over a router between Alice and Bob. She can see all messages that pass through this router and she can spoof messages from either Alice or Bob.
To beat this level, you need to trick Alice into encrypting her messages using a key Eve knows: **31337**. You must also trick Bob into thinking he is having a conversation with Alice as normal! See if you can figure out how to trick them both without them discovering your presence between them.
Hint: there is an example of how the encryption protocol works at the top of the field. Use the pause button to see what the values of the packet headers are in the protocol.
Level Objectives
trick Alice into encrypting her message with Eve's key: 31337
make sure Bob can still read the message, by encrypting it with his key

12


You win!
You won the level! Congrats!
Go to the next level
odin id :raghuram
Censorship
Level list
In this level, Alice's Internet browsing is being **censored.** She can browse an **allowed site**, but the censor will drop any packets to a **blocked site**.
One way to evade censorship is to make an intermediate connection to a **proxy server**. This server will take packets that are destined to it and redirect them to the blocked site. Since the censor can only see the link between Alice and the proxy, it does not know which site Alice is browsing through it.
To beat this level, you need to evade the censor and send a packet from Alice to the Blocked Site.
Level Objectives
Send a packet from Alice to Blocked Site.

13



01.3: Cloud networking

3. Scan targets for services

- Show a screenshot of the output for the scan for your lab notebook.

```
raghuram@course-vm:~$ nmap 10.138.0.5
Starting Nmap 7.80 ( https://nmap.org ) at 2024-04-09 00:47 UTC
Nmap scan report for joomla-1-vm.c.cloud-nataraja-raghuram.internal (10.138.0.5)
Host is up (0.00028s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https
```

5. Navigating default networks

- How many subnetworks are created initially on the default network? How many regions does this correspond to? (Use a pipe to pass output to grep in order to return specific lines of output and then another to pass output to wc to count them: | grep default | wc -l)

```
raghuram@cloudshell:~ (cloud-nataraja-raghuram)$ gcloud compute networks subnets list | grep NETWORK | wc -l
42
```

- Given the CIDR prefix associated with each subnetwork, how many hosts does each subnetwork support?

4094 hosts

- Which CIDR subnetworks are these instances brought up in? Do they correspond to the appropriate region based on the prior commands?

Yes they correspond to prior commands

Status	Name	Zone	Recommendations	In use by	Internal IP
✓	course-vm	us-west1-b			10.138.0.2 (nic0)
✓	instance-1	us-west1-b			10.138.0.6 (nic0)
✓	instance-2	us-west1-a			10.138.0.9 (nic0)

Related actions

odin id :raghuram

From instance-1, perform a ping to the Internal IP address of instance-2. Take a screenshot of the output.

```
raghuram@instance-1:~$ ping 10.138.0.9
PING 10.138.0.9 (10.138.0.9) 56(84) bytes of data.
```

- From the figure in the previous step. What facilitates this connectivity: the virtual switch or the VPN Gateway?

  virtual switch

6. Creating custom networks

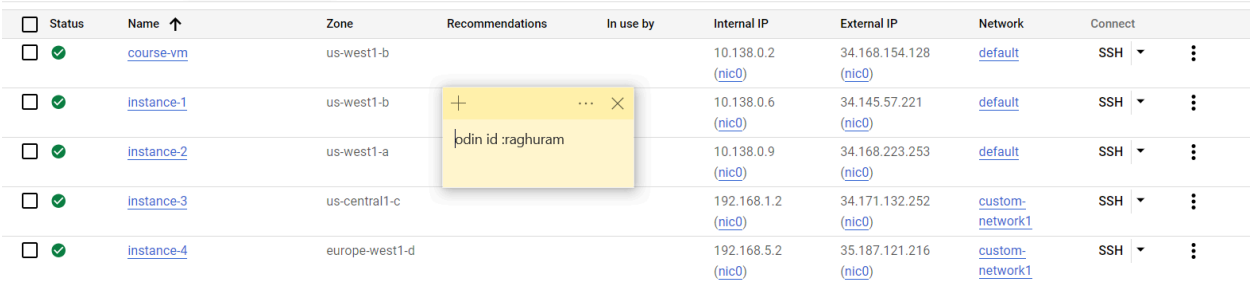
- Take a screenshot of the new subnets created in `custom-network1` alongside the default subnetworks in those regions assigned to the `default` network.

```
raghuram@cloudshell:~ (cloud-nataraja-raghuram)$ gcloud compute networks subnets list --regions=us-central1,europe-west1
NAME: default
REGION: europe-west1
NETWORK: default
RANGE: 10.132.0.0/20
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:

NAME: subnet-europe-west-192
REGION: europe-west1
NETWORK: custom-network1
RANGE: 192.168.5.0/24
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:

NAME: default
REGION: us-central1
NETWORK: default
RANGE: 10.128.0.0/20
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:

NAME: subnet-us-central-192
REGION: us-central1
NETWORK: custom-network1
RANGE: 192.168.1.0/24
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:
```

- Explain why the result of this ping is different from when you performed the ping to instance-2

The result is different as initially we had pinged from instance-1 to instance-2 which was on the same default network , when pinged from the instance-1 to instance-3 it was on the custom network

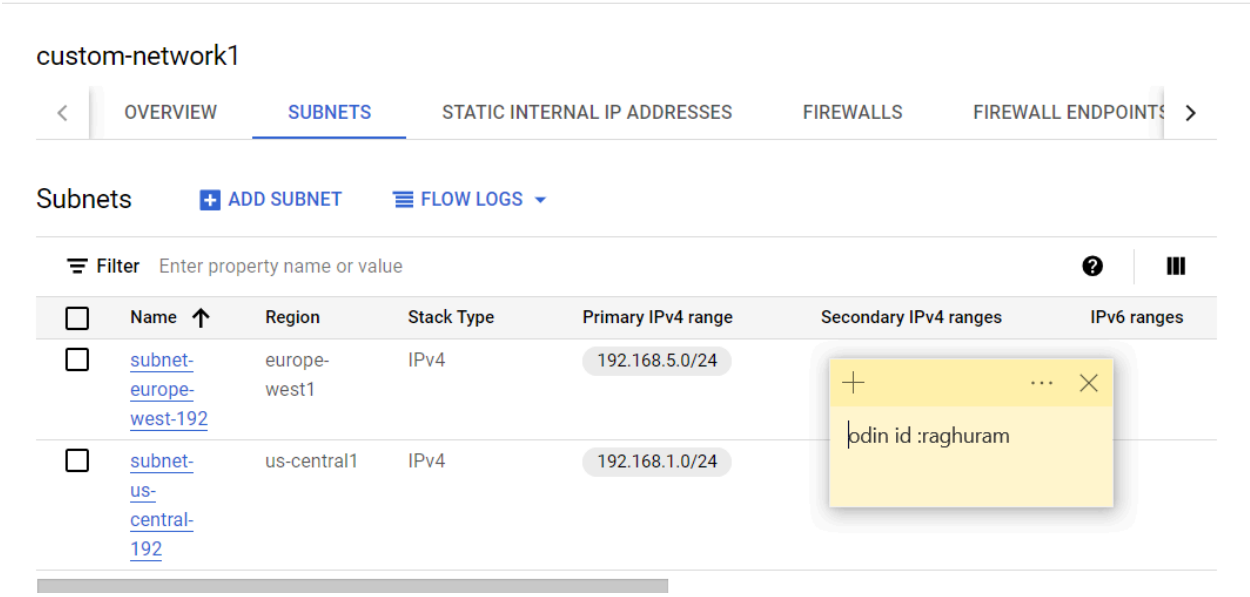
- Take screenshots of all 4 instances in the UI including the network they belong to.



Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Network	Connect
✓	course-vm	us-west1-b			10.138.0.2 (nic0)	34.168.154.128 (nic0)	default	SSH
✓	instance-1	us-west1-b			10.138.0.6 (nic0)	34.145.57.221 (nic0)	default	SSH
✓	instance-2	us-west1-a			10.138.0.9 (nic0)	34.168.223.253 (nic0)	default	SSH
✓	instance-3	us-central1-c			192.168.1.2 (nic0)	34.171.132.252 (nic0)	custom-network1	SSH
✓	instance-4	europe-west1-d			192.168.5.2 (nic0)	35.187.121.216 (nic0)	custom-network1	SSH

pdin id :raghuram

- Take a screenshot of the subnetworks created for the custom-network1 network and some of the subnetworks of the default network showing their regions, internal IP ranges and Gateways.



custom-network1

OVERVIEW | SUBNETS | STATIC INTERNAL IP ADDRESSES | FIREWALLS | FIREWALL ENDPOINTS

Subnets ADD SUBNET FLOW LOGS

Filter Enter property name or value

Name	Region	Stack Type	Primary IPv4 range	Secondary IPv4 ranges	IPv6 ranges
subnet-europe-west-192	europe-west1	IPv4	192.168.5.0/24		
subnet-us-central-192	us-central1	IPv4	192.168.1.0/24		

pdin id :raghuram

VPC network details

DELETE VPC NETWORK

Subnets

ADD SUBNET

FLOW LOGS

Filter Enter property name or value

Name	Region	Stack Type	Primary IPv4 range	Secondary IPv4 ranges	IPv6 ranges
default	us-central1	IPv4	10.128.0.0/20		
default	europe-west1	IPv4	10.132.0.0/20		
default	us-west1	IPv4	10.138.0.0/20		
default	asia-east1	IPv4	10.140.0.0/20		
default	us-east1	IPv4	10.142.0.0/20		
default	asia-northeast1	IPv4	10.146.0.0/20		
default	asia-southeast1	IPv4	10.148.0.0/20		
default	us-east4	IPv4	10.150.0.0/20		
default	australia-southeast1	IPv4	10.152.0.0/20		

odin id :raghuram