Project: Implementation of Virtual Customer Premise Equipment

0816034 蔡家倫

Part 1

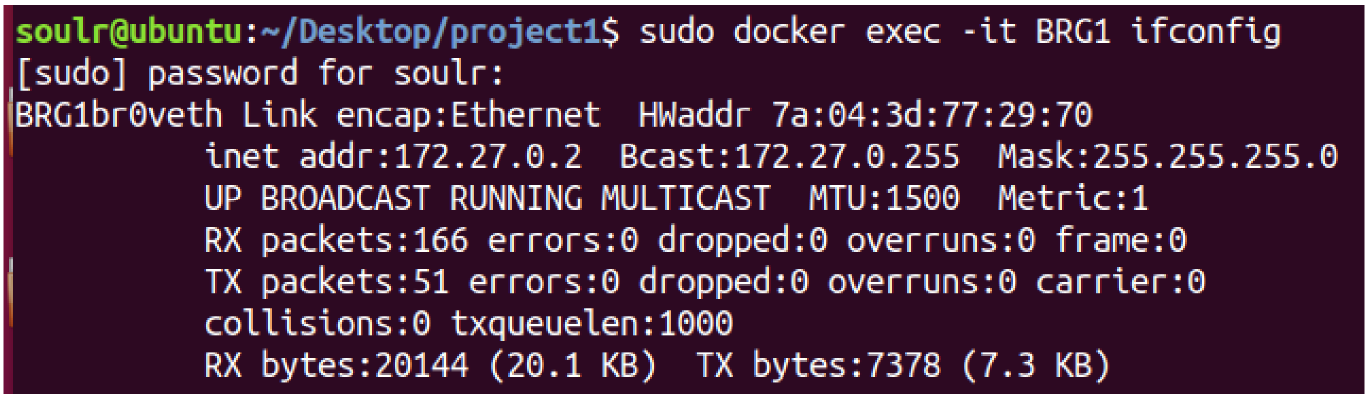
1. Node configurations (20%)

Show the configuration commands you made on each node to provide Internet connectivity for hosts and briefly explain the purpose of the commands (take screenshots to justify your answers)(10%)

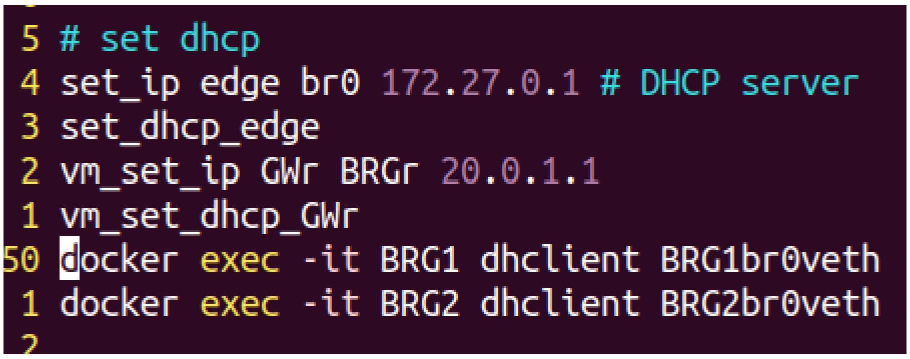
(a) DHCP on edge router (2%)

- BRG1 can acquire an IP address from Edge Router, respectively

I run "dhclient" command to get IP from DHCP server on **edge**.



```
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it BRG1 ifconfig
[sudo] password for soulr:
BRG1br0veth Link encap:Ethernet  HWaddr 7a:04:3d:77:29:70
          inet addr:172.27.0.2  Bcast:172.27.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:166 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20144 (20.1 KB)  TX bytes:7378 (7.3 KB)
```

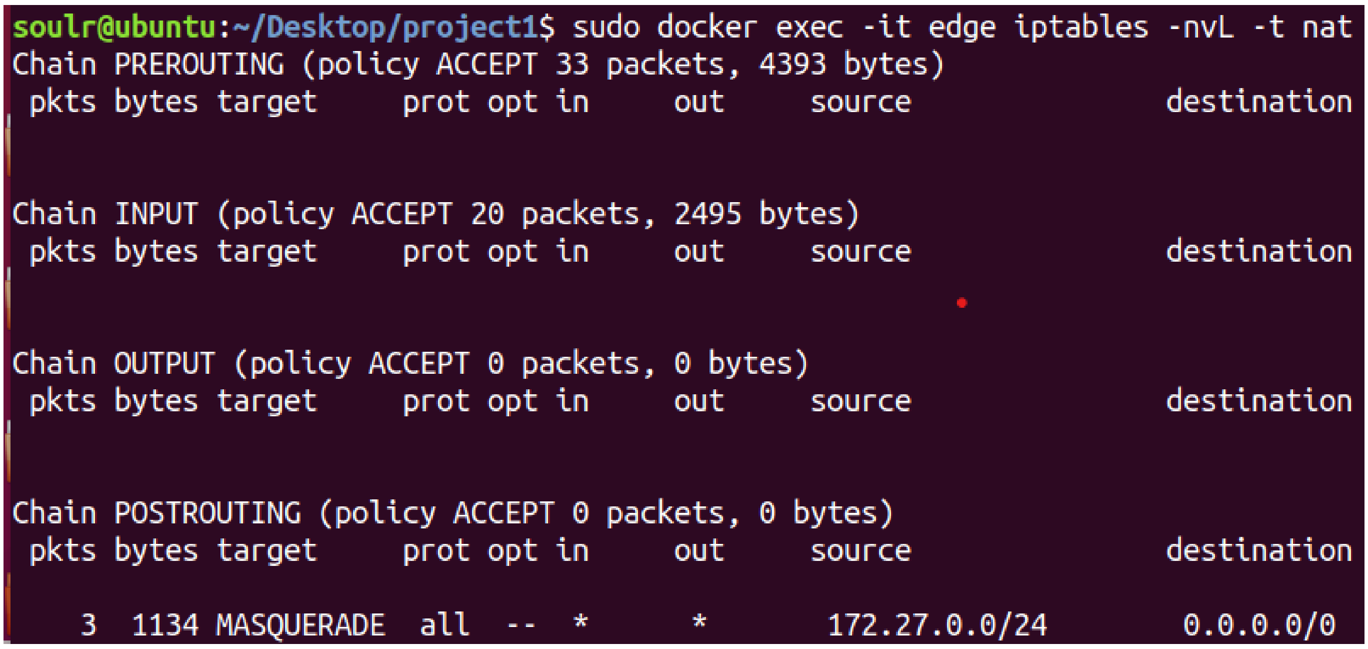


```
# set dhcp
set_ip edge br0 172.27.0.1 # DHCP server
set_dhcp_edge
vm_set_ip GWr BRGr 20.0.1.1
vm_set_dhcp_GWr
docker exec -it BRG1 dhclient BRG1br0veth
docker exec -it BRG2 dhclient BRG2br0veth
```

(b) NAT on Edge Router (5%)

- Use iptables command to show NAT rules

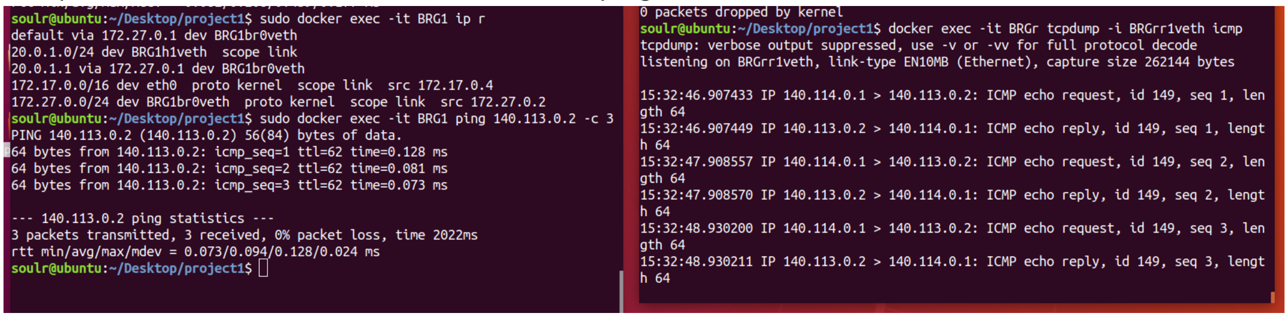
I use "MASQUERADE" to change source IP of the packet.



```
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it edge iptables -nvL -t nat
Chain PREROUTING (policy ACCEPT 33 packets, 4393 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain INPUT (policy ACCEPT 20 packets, 2495 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    3  1134 MASQUERADE  all  --  *      *       172.27.0.0/24        0.0.0.0/0
```

(c) GRE over UDP (5%)

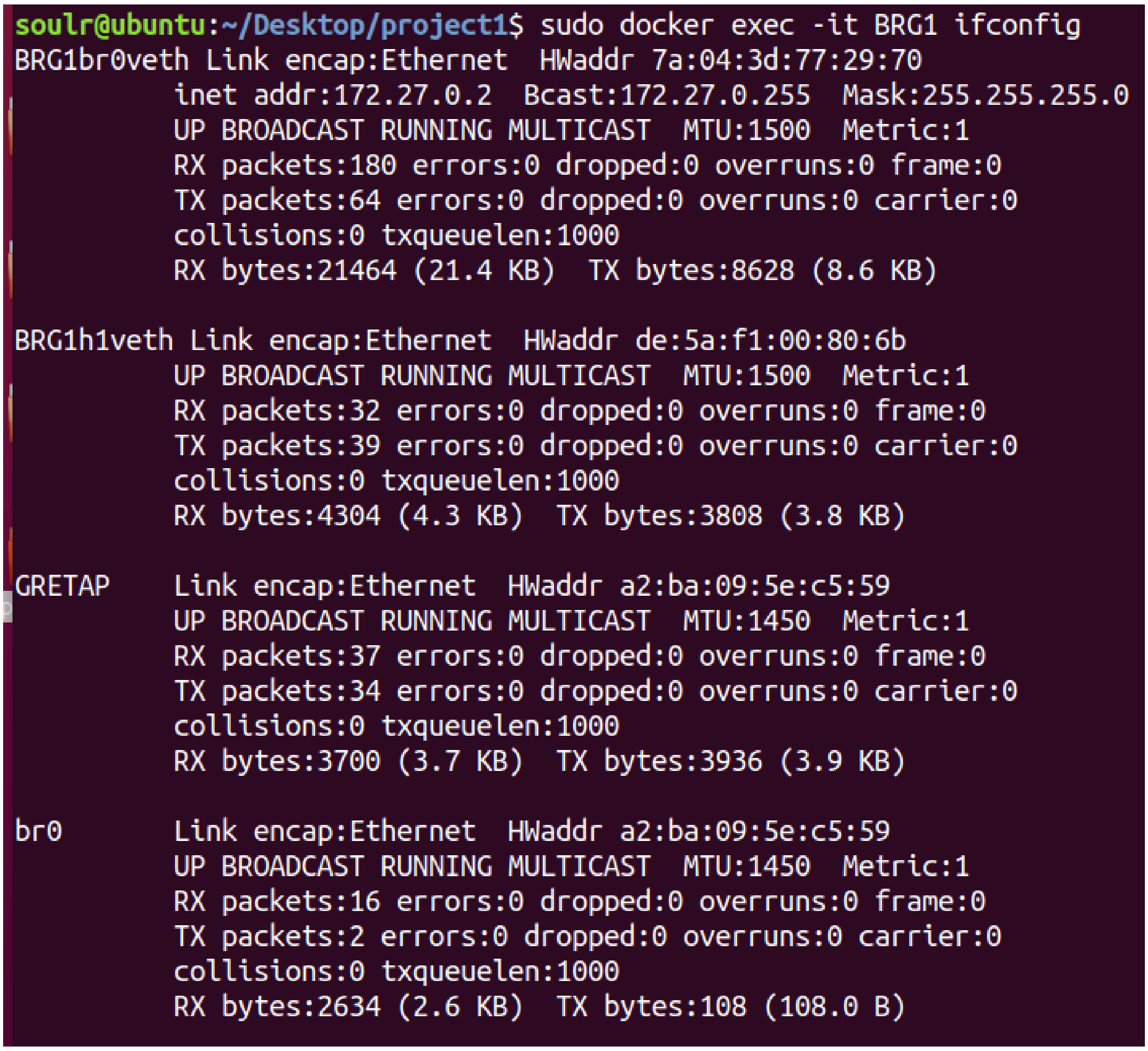
- Setup BRG1, but not BRG2. Show that BRG1 can ping BRGr.



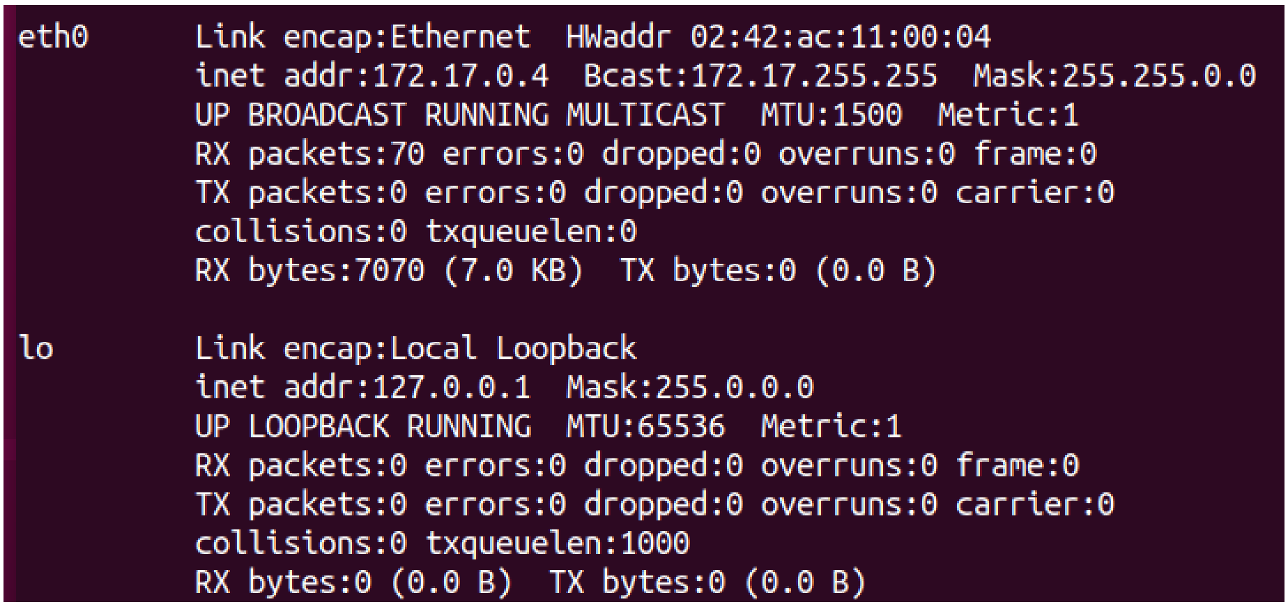
```
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it BRG1 ip r
default via 172.27.0.1 dev BRG1br0veth
20.0.1.0/24 dev BRG1h1veth  scope link
20.0.1.1 via 172.27.0.1 dev BRG1br0veth
172.17.0.0/16 dev eth0  proto kernel  scope link  src 172.17.0.4
172.27.0.0/24 dev BRG1br0veth  proto kernel  scope link  src 172.27.0.2
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it BRG1 ping 140.113.0.2 -c 3
PING 140.113.0.2 (140.113.0.2) 56(84) bytes of data.
64 bytes from 140.113.0.2: icmp_seq=1 ttl=62 time=0.128 ms
64 bytes from 140.113.0.2: icmp_seq=2 ttl=62 time=0.081 ms
64 bytes from 140.113.0.2: icmp_seq=3 ttl=62 time=0.073 ms

--- 140.113.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2022ms
rtt min/avg/max/mdev = 0.073/0.094/0.128/0.024 ms
soulr@ubuntu:~/Desktop/project1$ 
```

```
0 packets dropped by kernel
soulr@ubuntu:~/Desktop/project1$ docker exec -it BRGr tcpdump -i BRGrr1veth icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on BRGrr1veth, link-type EN10MB (Ethernet), capture size 262144 bytes

15:32:46.907433 IP 140.114.0.1 > 140.113.0.2: ICMP echo request, id 149, seq 1, len
gth 64
15:32:46.907449 IP 140.113.0.2 > 140.114.0.1: ICMP echo reply, id 149, seq 1, lengt
h 64
15:32:47.908557 IP 140.114.0.1 > 140.113.0.2: ICMP echo request, id 149, seq 2, len
gth 64
15:32:47.908570 IP 140.113.0.2 > 140.114.0.1: ICMP echo reply, id 149, seq 2, lengt
h 64
15:32:48.930200 IP 140.114.0.1 > 140.113.0.2: ICMP echo request, id 149, seq 3, len
gth 64
15:32:48.930211 IP 140.113.0.2 > 140.114.0.1: ICMP echo reply, id 149, seq 3, lengt
h 64
```

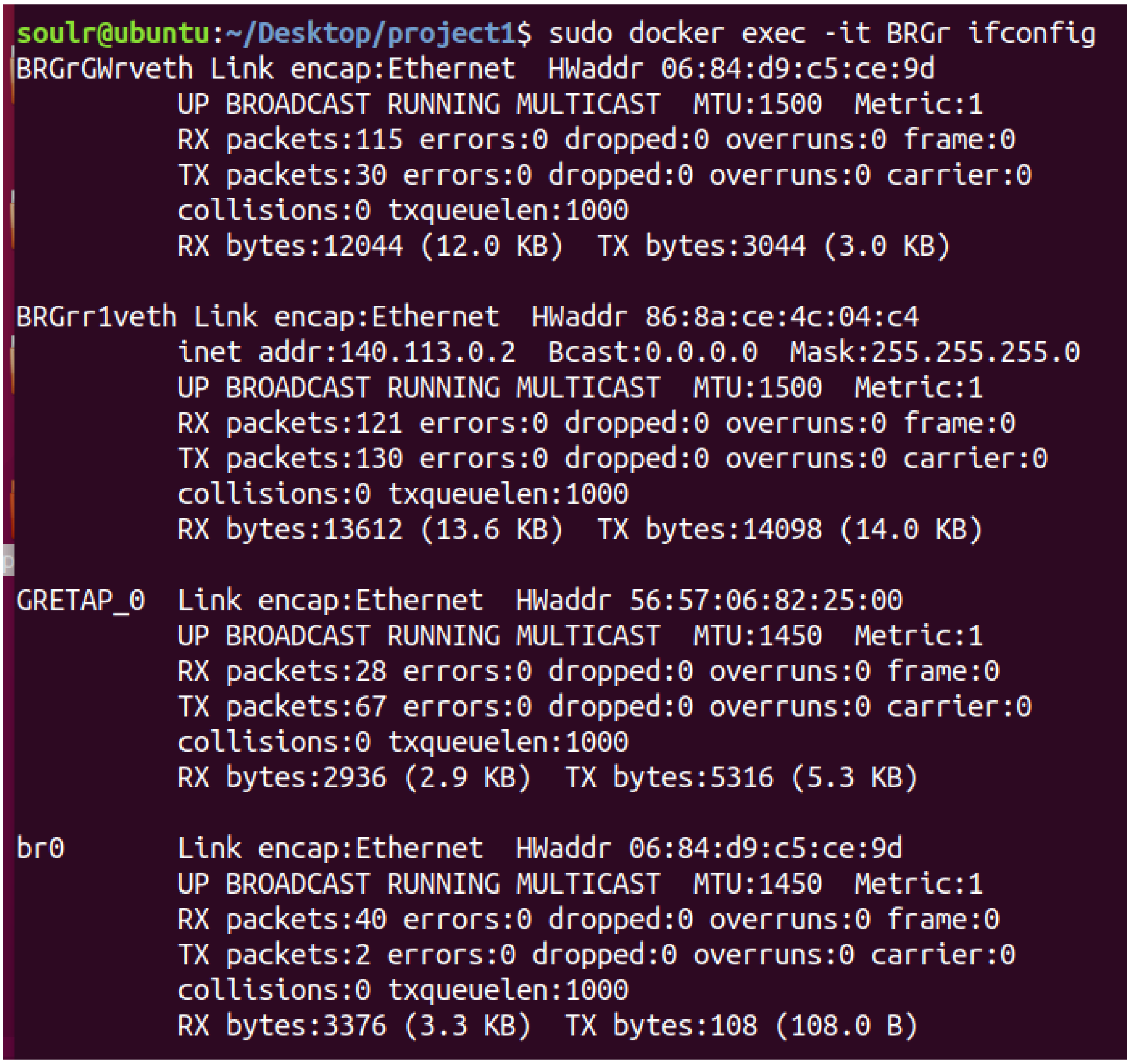
- Show interfaces list on node BRG1 and BRGr



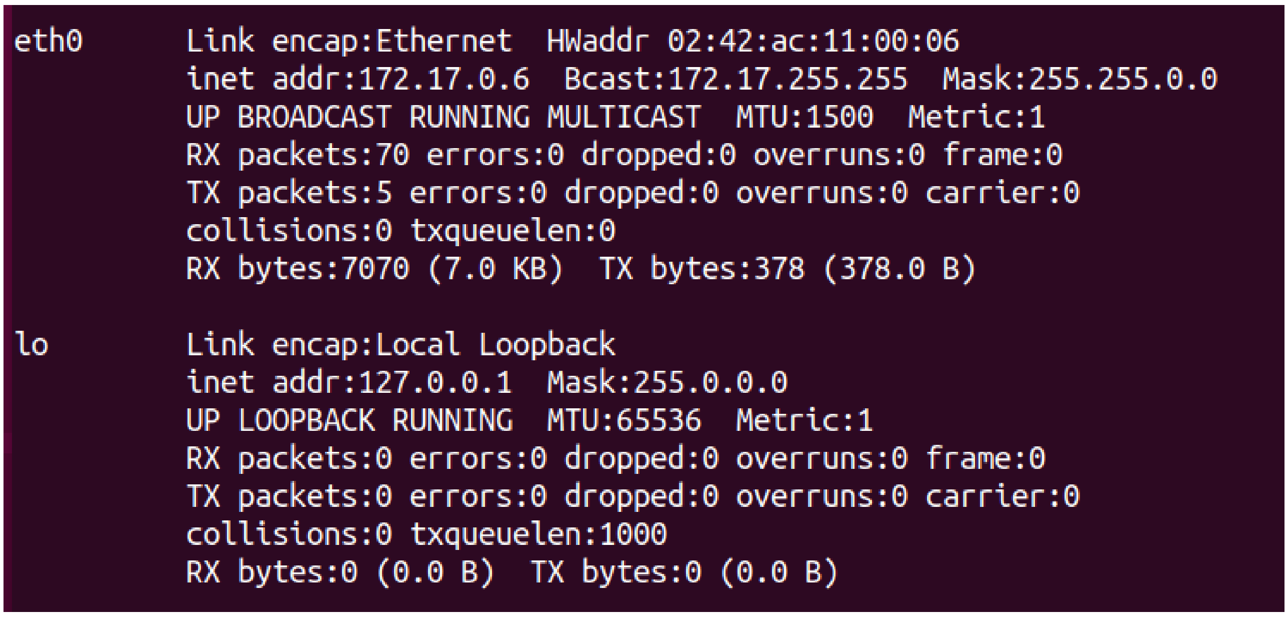
```
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it BRG1 ifconfig
BRG1br0veth Link encap:Ethernet  HWaddr 7a:04:3d:77:29:70
          inet addr:172.27.0.2  Bcast:172.27.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:180 errors:0 dropped:0 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21464 (21.4 KB)  TX bytes:8628 (8.6 KB)

BRG1h1veth Link encap:Ethernet  HWaddr de:5a:f1:00:80:6b
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4304 (4.3 KB)  TX bytes:3808 (3.8 KB)

GRETAP    Link encap:Ethernet  HWaddr a2:ba:09:5e:c5:59
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3700 (3.7 KB)  TX bytes:3936 (3.9 KB)

br0       Link encap:Ethernet  HWaddr a2:ba:09:5e:c5:59
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2634 (2.6 KB)  TX bytes:108 (108.0 B)
```



```
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:04
          inet addr:172.17.0.4  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7070 (7.0 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```



```
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it BRGr ifconfig
BRGrGWrveth Link encap:Ethernet  HWaddr 06:84:d9:c5:ce:9d
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12044 (12.0 KB)  TX bytes:3044 (3.0 KB)

BRGrr1veth Link encap:Ethernet  HWaddr 86:8a:ce:4c:04:c4
          inet addr:140.113.0.2  Bcast:0.0.0.0  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:121 errors:0 dropped:0 overruns:0 frame:0
          TX packets:130 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13612 (13.6 KB)  TX bytes:14098 (14.0 KB)

GRETAP_0  Link encap:Ethernet  HWaddr 56:57:06:82:25:00
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:28 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2936 (2.9 KB)  TX bytes:5316 (5.3 KB)

br0       Link encap:Ethernet  HWaddr 06:84:d9:c5:ce:9d
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:40 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3376 (3.3 KB)  TX bytes:108 (108.0 B)
```

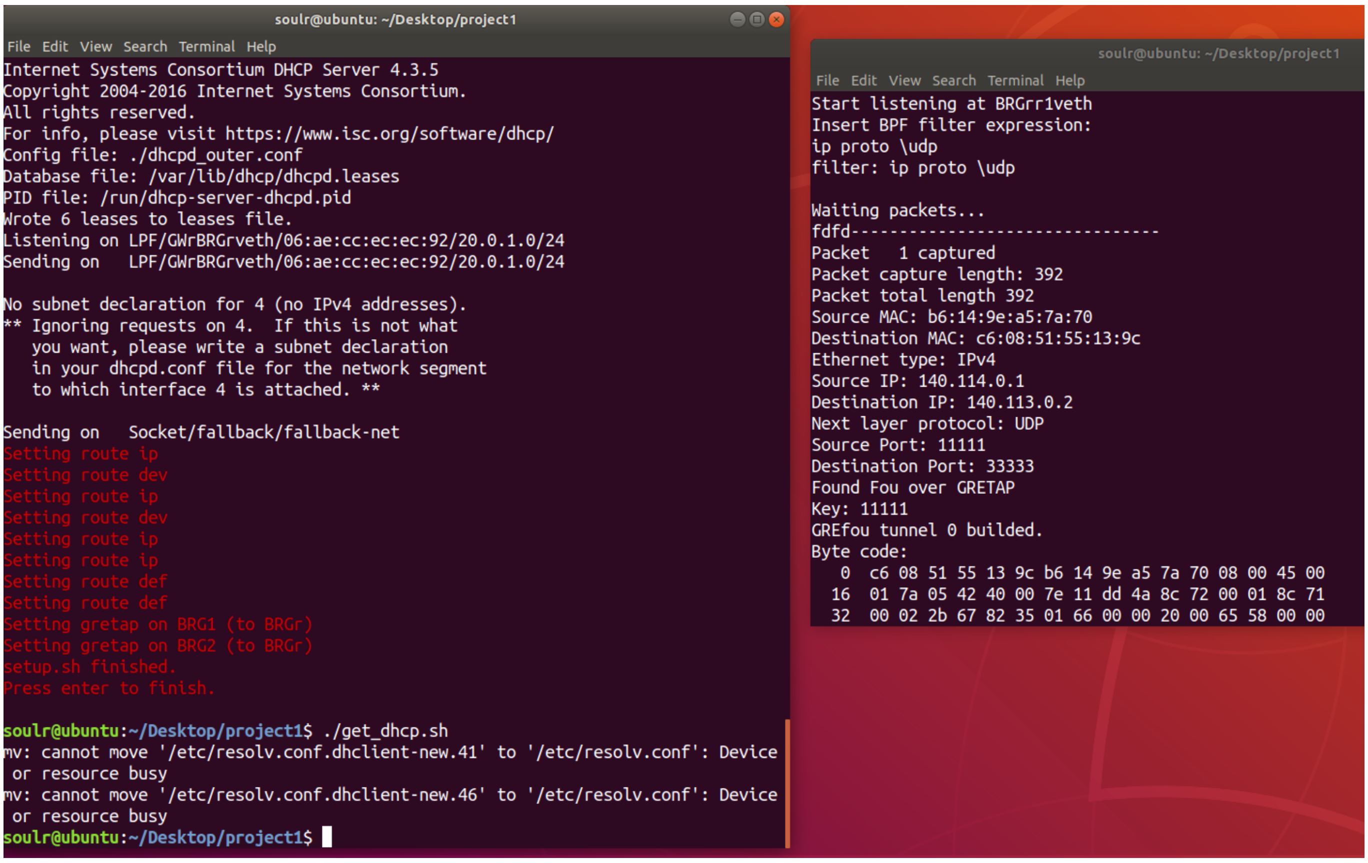


```
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:06
          inet addr:172.17.0.6  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7070 (7.0 KB)  TX bytes:378 (378.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

(d) DHCP on GWr (2%)

- H1 acquires IP and gateway dynamically from DHCP on GWr

I run **setup.sh** to do basic setup, run **run_pcap.sh** to capture packet and build GRETAP tunnel (right pane), and run **get_dhcp.sh** to run dhcp command.

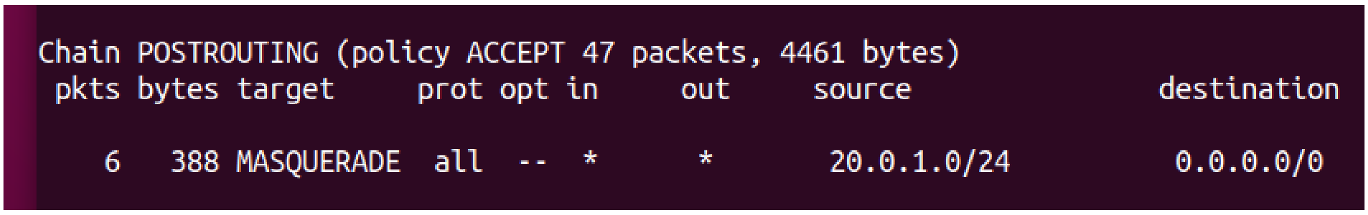


```
Internet Systems Consortium DHCP Server 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: ./dhcpd_outer.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /run/dhcp-server-dhcpd.pid
Wrote 6 leases to leases file.
Listening on LPF/GWrBRGrveth/06:ae:cc:ec:ec:92/20.0.1.0/24
Sending on   LPF/GWrBRGrveth/06:ae:cc:ec:ec:92/20.0.1.0/24

No subnet declaration for 4 (no IPv4 addresses).
** Ignoring requests on 4.  If this is not what
   you want, please write a subnet declaration
   in your dhcpd.conf file for the network segment
   to which interface 4 is attached. **

Sending on   Socket/fallback/fallback-net
Setting route ip
Setting route dev
Setting route ip
Setting route dev
Setting route ip
Setting route ip
Setting route def
Setting route def
Setting gretap on BRG1 (to BRGr)
Setting gretap on BRG2 (to BRGr)
setup.sh finished.
Press enter to finish.

soulr@ubuntu:~/Desktop/project1$ ./get_dhcp.sh
mv: cannot move '/etc/resolv.conf.dhclient-new.41' to '/etc/resolv.conf': Device
 or resource busy
mv: cannot move '/etc/resolv.conf.dhclient-new.46' to '/etc/resolv.conf': Device
 or resource busy
soulr@ubuntu:~/Desktop/project1$
```

```
Start listening at BRGrr1veth
Insert BPF filter expression:
ip proto \udp
filter: ip proto \udp

Waiting packets...
fdfd--------------------------------
Packet   1 captured
Packet capture length: 392
Packet total length 392
Source MAC: b6:14:9e:a5:7a:70
Destination MAC: c6:08:51:55:13:9c
Ethernet type: IPv4
Source IP: 140.114.0.1
Destination IP: 140.113.0.2
Next layer protocol: UDP
Source Port: 11111
Destination Port: 33333
Found Fou over GRETAP
Key: 11111
GREfou tunnel 0 builded.
Byte code:
   0  c6 08 51 55 13 9c b6 14 9e a5 7a 70 08 00 45 00
  16  01 7a 05 42 40 00 7e 11 dd 4a 8c 72 00 01 8c 71
  32  00 02 2b 67 82 35 01 66 00 00 20 00 65 58 00 00
```

(e) NAT on GWr

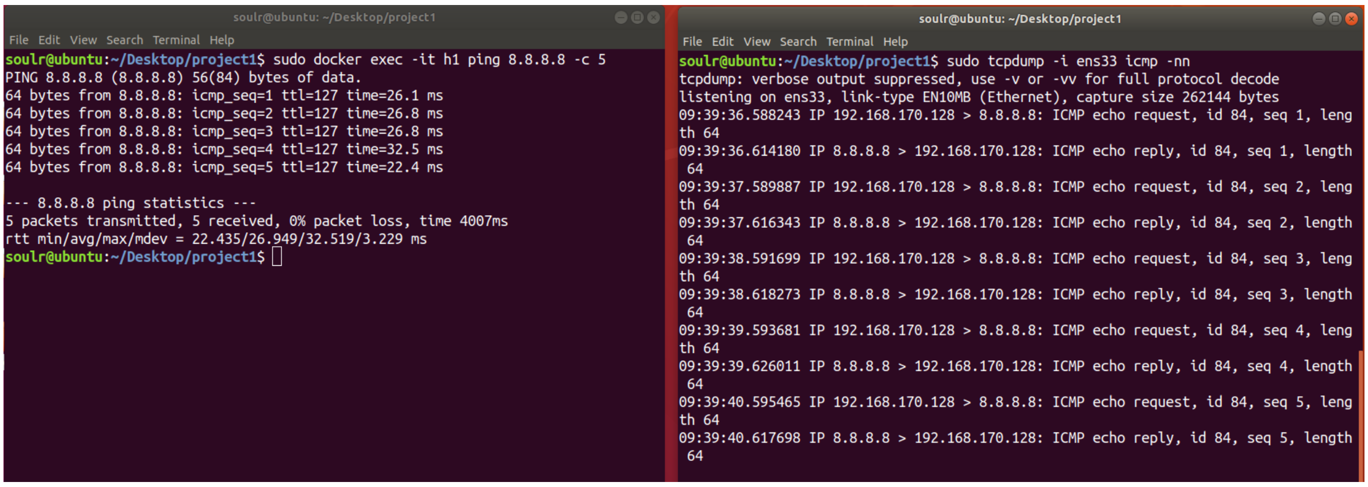
- Use iptables command to show NAT rules



```
Chain POSTROUTING (policy ACCEPT 47 packets, 4461 bytes)
 pkts bytes target     prot opt in     out     source               destination

    6   388 MASQUERADE  all  --  *      *       20.0.1.0/24          0.0.0.0/0
```

(f) Ping to internet

- Show that H1 can ping google DNS Server 8.8.8.8

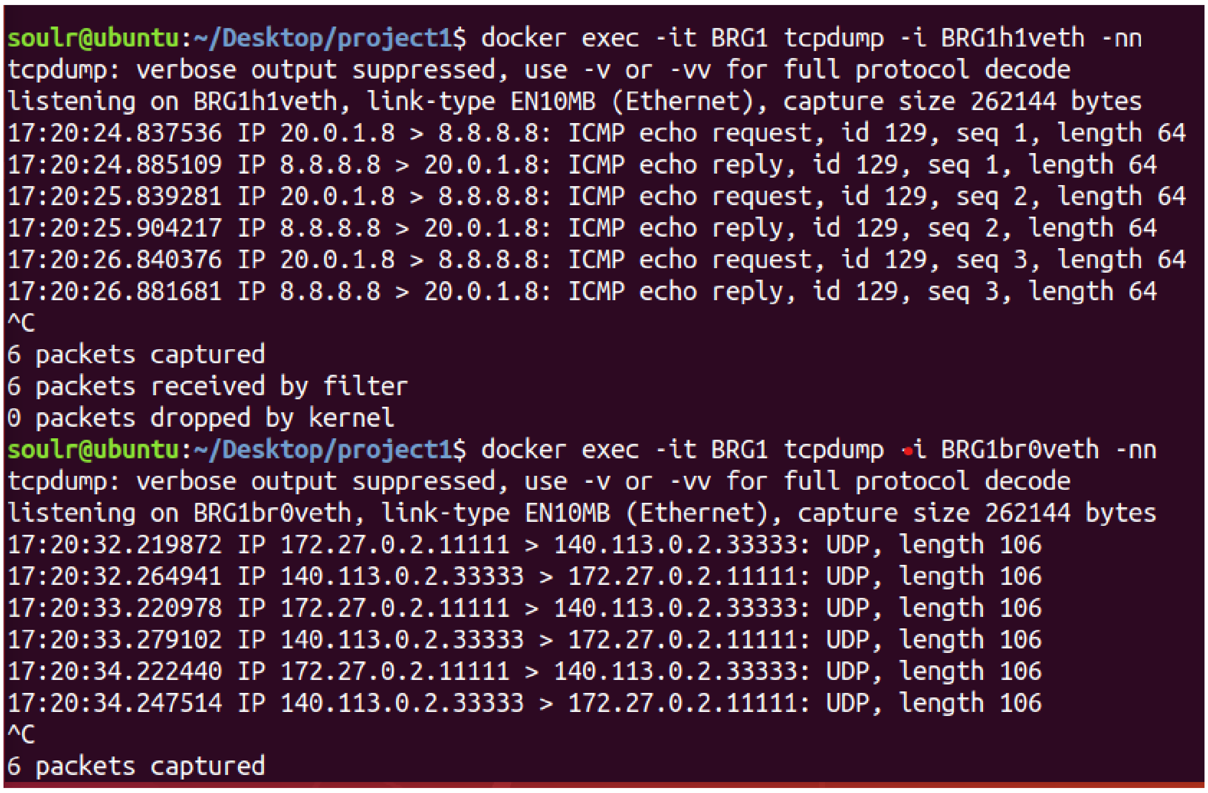


```
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it h1 ping 8.8.8.8 -c 5
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=26.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=26.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=127 time=26.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=127 time=32.5 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=127 time=22.4 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 22.435/26.949/32.519/3.229 ms
soulr@ubuntu:~/Desktop/project1$
```

```
soulr@ubuntu:~/Desktop/project1$ sudo tcpdump -i ens33 icmp -nn
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
09:39:36.588243 IP 192.168.170.128 > 8.8.8.8: ICMP echo request, id 84, seq 1, leng
th 64
09:39:36.614180 IP 8.8.8.8 > 192.168.170.128: ICMP echo reply, id 84, seq 1, length
 64
09:39:37.589887 IP 192.168.170.128 > 8.8.8.8: ICMP echo request, id 84, seq 2, leng
th 64
09:39:37.616343 IP 8.8.8.8 > 192.168.170.128: ICMP echo reply, id 84, seq 2, length
 64
09:39:38.591699 IP 192.168.170.128 > 8.8.8.8: ICMP echo request, id 84, seq 3, leng
th 64
09:39:38.618273 IP 8.8.8.8 > 192.168.170.128: ICMP echo reply, id 84, seq 3, length
 64
09:39:39.593681 IP 192.168.170.128 > 8.8.8.8: ICMP echo request, id 84, seq 4, leng
th 64
09:39:39.626011 IP 8.8.8.8 > 192.168.170.128: ICMP echo reply, id 84, seq 4, length
 64
09:39:40.595465 IP 192.168.170.128 > 8.8.8.8: ICMP echo request, id 84, seq 5, leng
th 64
09:39:40.617698 IP 8.8.8.8 > 192.168.170.128: ICMP echo reply, id 84, seq 5, length
 64
```

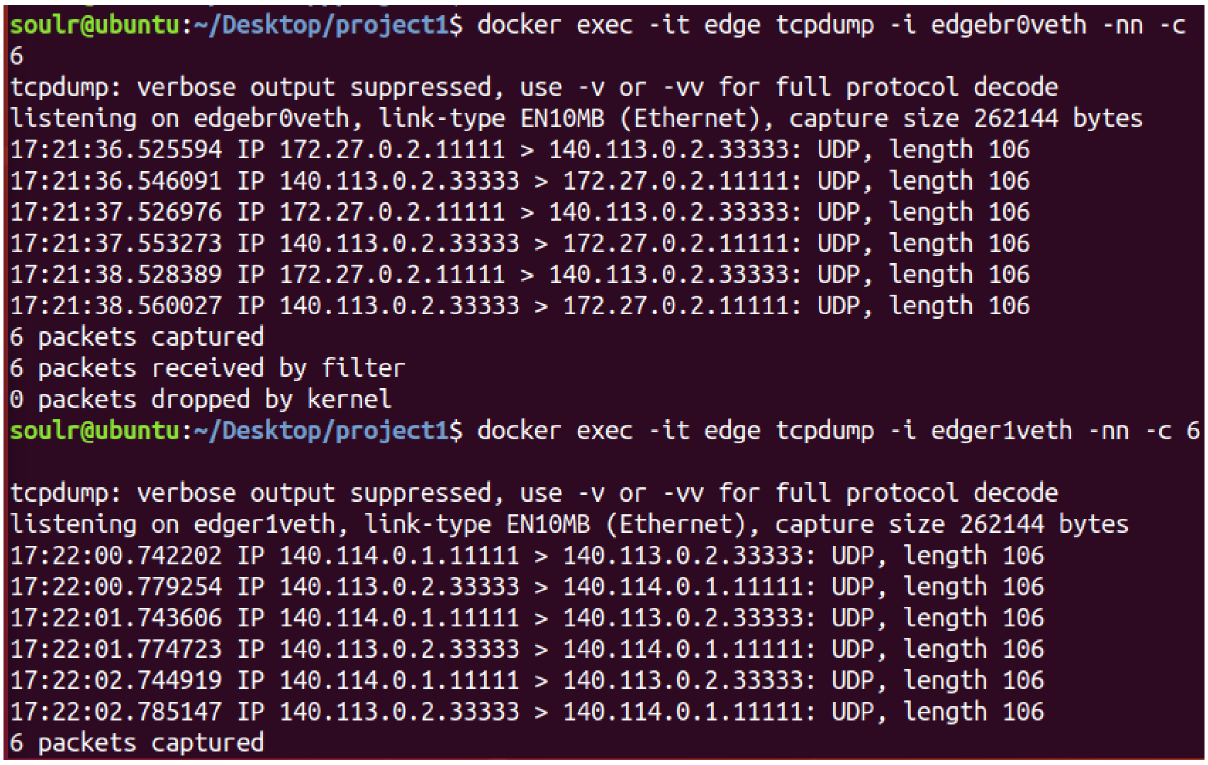
2. Route Trace (5%)

- Let H1 ping 8.8.8.8. Capture packets and take screenshots on nodes and briefly describe the change in packet headers
- BRG1, BRGr, GWr, Edge Router input/output
  - BRG1



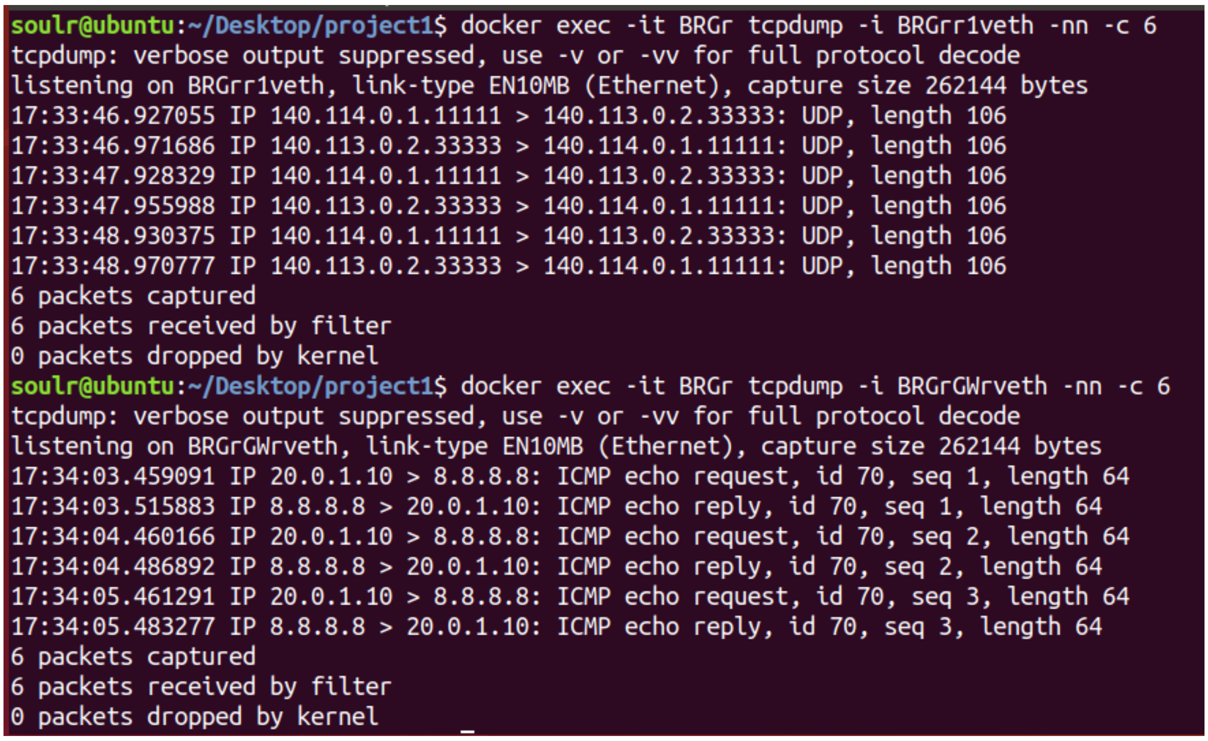
```
soulr@ubuntu:~/Desktop/project1$ docker exec -it BRG1 tcpdump -i BRG1h1veth -nn
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on BRG1h1veth, link-type EN10MB (Ethernet), capture size 262144 bytes
17:20:24.837536 IP 20.0.1.8 > 8.8.8.8: ICMP echo request, id 129, seq 1, length 64
17:20:24.885109 IP 8.8.8.8 > 20.0.1.8: ICMP echo reply, id 129, seq 1, length 64
17:20:25.839281 IP 20.0.1.8 > 8.8.8.8: ICMP echo request, id 129, seq 2, length 64
17:20:25.904217 IP 8.8.8.8 > 20.0.1.8: ICMP echo reply, id 129, seq 2, length 64
17:20:26.840376 IP 20.0.1.8 > 8.8.8.8: ICMP echo request, id 129, seq 3, length 64
17:20:26.881681 IP 8.8.8.8 > 20.0.1.8: ICMP echo reply, id 129, seq 3, length 64
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
soulr@ubuntu:~/Desktop/project1$ docker exec -it BRG1 tcpdump -i BRG1br0veth -nn
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on BRG1br0veth, link-type EN10MB (Ethernet), capture size 262144 bytes
17:20:32.219872 IP 172.27.0.2.11111 > 140.113.0.2.33333: UDP, length 106
17:20:32.264941 IP 140.113.0.2.33333 > 172.27.0.2.11111: UDP, length 106
17:20:33.220978 IP 172.27.0.2.11111 > 140.113.0.2.33333: UDP, length 106
17:20:33.279102 IP 140.113.0.2.33333 > 172.27.0.2.11111: UDP, length 106
17:20:34.222440 IP 172.27.0.2.11111 > 140.113.0.2.33333: UDP, length 106
17:20:34.247514 IP 140.113.0.2.33333 > 172.27.0.2.11111: UDP, length 106
^C
6 packets captured
```

  - edge



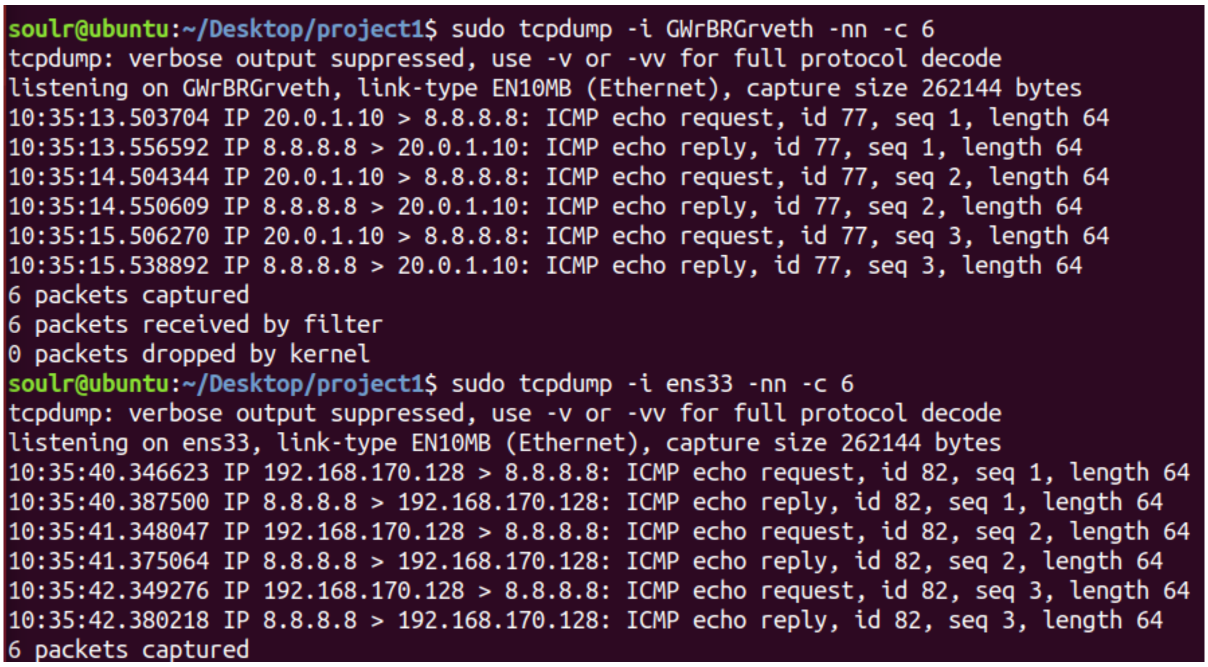
```
soulr@ubuntu:~/Desktop/project1$ docker exec -it edge tcpdump -i edgebr0veth -nn -c
6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on edgebr0veth, link-type EN10MB (Ethernet), capture size 262144 bytes
17:21:36.525594 IP 172.27.0.2.11111 > 140.113.0.2.33333: UDP, length 106
17:21:36.546091 IP 140.113.0.2.33333 > 172.27.0.2.11111: UDP, length 106
17:21:37.526976 IP 172.27.0.2.11111 > 140.113.0.2.33333: UDP, length 106
17:21:37.553273 IP 140.113.0.2.33333 > 172.27.0.2.11111: UDP, length 106
17:21:38.528389 IP 172.27.0.2.11111 > 140.113.0.2.33333: UDP, length 106
17:21:38.560027 IP 140.113.0.2.33333 > 172.27.0.2.11111: UDP, length 106
6 packets captured
6 packets received by filter
0 packets dropped by kernel
soulr@ubuntu:~/Desktop/project1$ docker exec -it edge tcpdump -i edger1veth -nn -c 6

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on edger1veth, link-type EN10MB (Ethernet), capture size 262144 bytes
17:22:00.742202 IP 140.114.0.1.11111 > 140.113.0.2.33333: UDP, length 106
17:22:00.779254 IP 140.113.0.2.33333 > 140.114.0.1.11111: UDP, length 106
17:22:01.743606 IP 140.114.0.1.11111 > 140.113.0.2.33333: UDP, length 106
17:22:01.774723 IP 140.113.0.2.33333 > 140.114.0.1.11111: UDP, length 106
17:22:02.744919 IP 140.114.0.1.11111 > 140.113.0.2.33333: UDP, length 106
17:22:02.785147 IP 140.113.0.2.33333 > 140.114.0.1.11111: UDP, length 106
6 packets captured
```

- BRGr



```
soulr@ubuntu:~/Desktop/project1$ docker exec -it BRGr tcpdump -i BRGrr1veth -nn -c 6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on BRGrr1veth, link-type EN10MB (Ethernet), capture size 262144 bytes
17:33:46.927055 IP 140.114.0.1.11111 > 140.113.0.2.33333: UDP, length 106
17:33:46.971686 IP 140.113.0.2.33333 > 140.114.0.1.11111: UDP, length 106
17:33:47.928329 IP 140.114.0.1.11111 > 140.113.0.2.33333: UDP, length 106
17:33:47.955988 IP 140.113.0.2.33333 > 140.114.0.1.11111: UDP, length 106
17:33:48.930375 IP 140.114.0.1.11111 > 140.113.0.2.33333: UDP, length 106
17:33:48.970777 IP 140.113.0.2.33333 > 140.114.0.1.11111: UDP, length 106
6 packets captured
6 packets received by filter
0 packets dropped by kernel
soulr@ubuntu:~/Desktop/project1$ docker exec -it BRGr tcpdump -i BRGrGWrveth -nn -c 6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on BRGrGWrveth, link-type EN10MB (Ethernet), capture size 262144 bytes
17:34:03.459091 IP 20.0.1.10 > 8.8.8.8: ICMP echo request, id 70, seq 1, length 64
17:34:03.515883 IP 8.8.8.8 > 20.0.1.10: ICMP echo reply, id 70, seq 1, length 64
17:34:04.460166 IP 20.0.1.10 > 8.8.8.8: ICMP echo request, id 70, seq 2, length 64
17:34:04.486892 IP 8.8.8.8 > 20.0.1.10: ICMP echo reply, id 70, seq 2, length 64
17:34:05.461291 IP 20.0.1.10 > 8.8.8.8: ICMP echo request, id 70, seq 3, length 64
17:34:05.483277 IP 8.8.8.8 > 20.0.1.10: ICMP echo reply, id 70, seq 3, length 64
6 packets captured
6 packets received by filter
0 packets dropped by kernel
```

- GWr



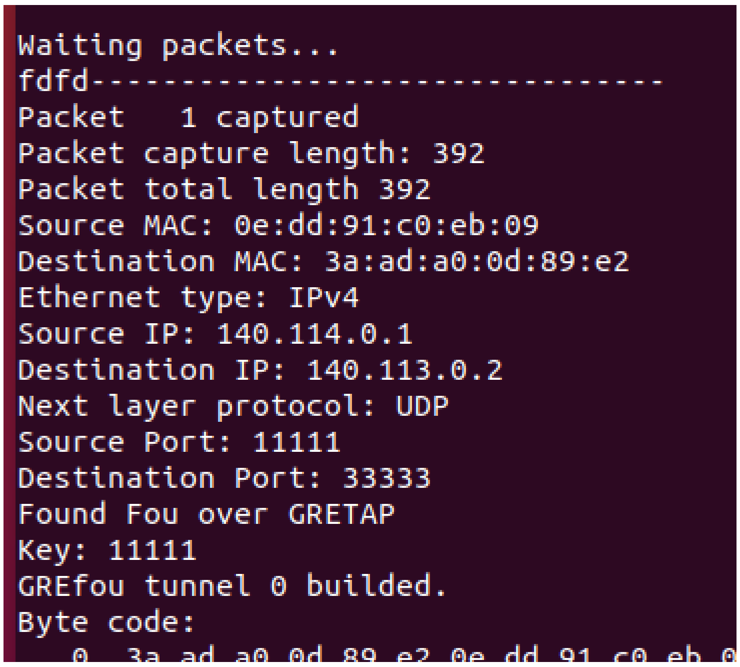
```
soulr@ubuntu:~/Desktop/project1$ sudo tcpdump -i GWrBRGrveth -nn -c 6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on GWrBRGrveth, link-type EN10MB (Ethernet), capture size 262144 bytes
10:35:13.503704 IP 20.0.1.10 > 8.8.8.8: ICMP echo request, id 77, seq 1, length 64
10:35:13.556592 IP 8.8.8.8 > 20.0.1.10: ICMP echo reply, id 77, seq 1, length 64
10:35:14.504344 IP 20.0.1.10 > 8.8.8.8: ICMP echo request, id 77, seq 2, length 64
10:35:14.550609 IP 8.8.8.8 > 20.0.1.10: ICMP echo reply, id 77, seq 2, length 64
10:35:15.506270 IP 20.0.1.10 > 8.8.8.8: ICMP echo request, id 77, seq 3, length 64
10:35:15.538892 IP 8.8.8.8 > 20.0.1.10: ICMP echo reply, id 77, seq 3, length 64
6 packets captured
6 packets received by filter
0 packets dropped by kernel
soulr@ubuntu:~/Desktop/project1$ sudo tcpdump -i ens33 -nn -c 6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
10:35:40.346623 IP 192.168.170.128 > 8.8.8.8: ICMP echo request, id 82, seq 1, length 64
10:35:40.387500 IP 8.8.8.8 > 192.168.170.128: ICMP echo reply, id 82, seq 1, length 64
10:35:41.348047 IP 192.168.170.128 > 8.8.8.8: ICMP echo request, id 82, seq 2, length 64
10:35:41.375064 IP 8.8.8.8 > 192.168.170.128: ICMP echo reply, id 82, seq 2, length 64
10:35:42.349276 IP 192.168.170.128 > 8.8.8.8: ICMP echo request, id 82, seq 3, length 64
10:35:42.380218 IP 8.8.8.8 > 192.168.170.128: ICMP echo reply, id 82, seq 3, length 64
6 packets captured
```

- Briefly explain the purposes of the header changes
  - BRG1: GRETAP tunnel with FOU
  - edge: NAT (內網 ip 轉 public ip)
  - BRGr: decap GRETAP packet
  - GWr: NAT

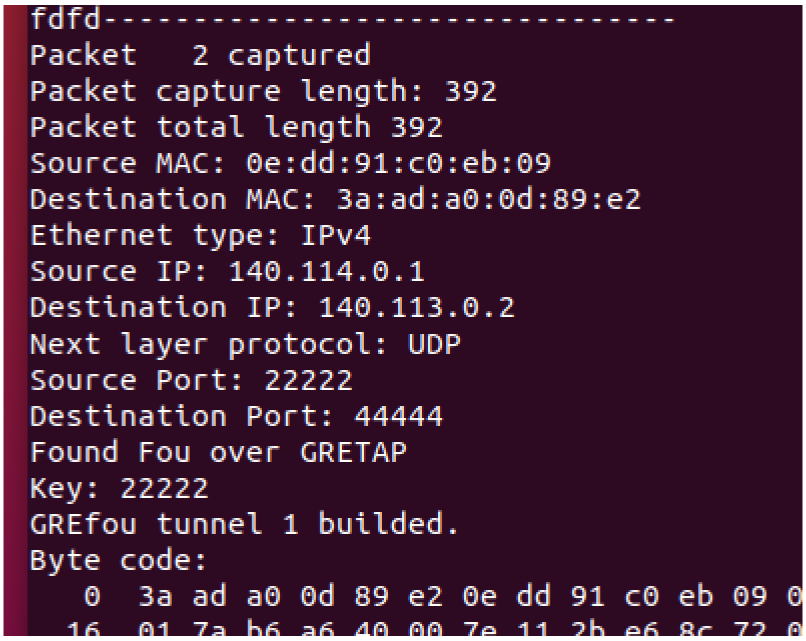
3. Setup both BRG1 and BRG2. (5%)

(a) Show that both BRG1 and BRG2 can create GRETAP tunnels with BRGr. Explain how BRGr distinguishes the two GRETAP tunnels

BRGr distinguishes 2 GRETAP tunnels by GRE key.



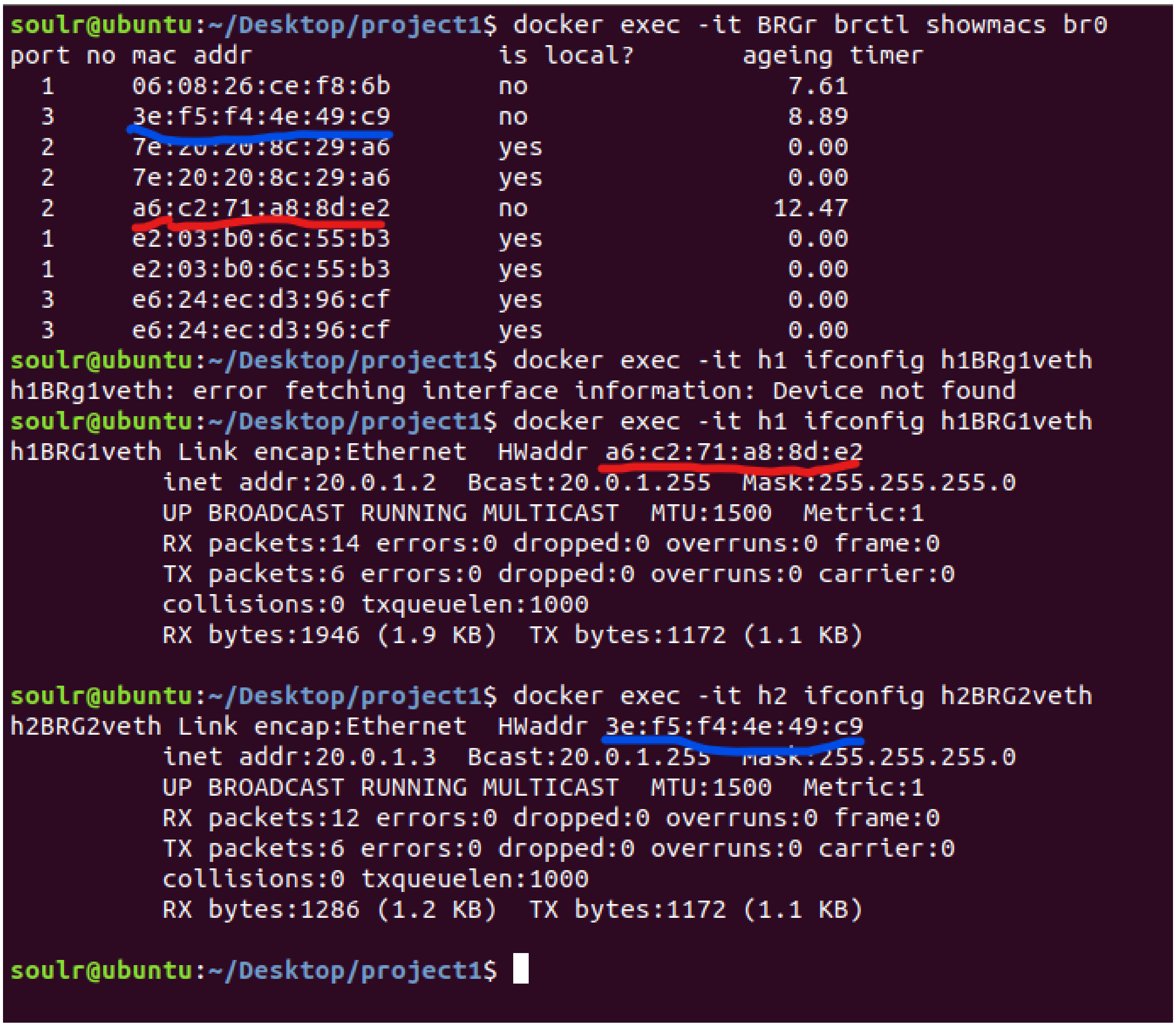
```
Waiting packets...
fdfd--------------------------------
Packet   1 captured
Packet capture length: 392
Packet total length 392
Source MAC: 0e:dd:91:c0:eb:09
Destination MAC: 3a:ad:a0:0d:89:e2
Ethernet type: IPv4
Source IP: 140.114.0.1
Destination IP: 140.113.0.2
Next layer protocol: UDP
Source Port: 11111
Destination Port: 33333
Found Fou over GRETAP
Key: 11111
GREfou tunnel 0 builded.
Byte code:
   0  3a ad a0 0d 89 e2 0e dd 91 c0 eb 0
```



```
fdfd--------------------------------
Packet   2 captured
Packet capture length: 392
Packet total length 392
Source MAC: 0e:dd:91:c0:eb:09
Destination MAC: 3a:ad:a0:0d:89:e2
Ethernet type: IPv4
Source IP: 140.114.0.1
Destination IP: 140.113.0.2
Next layer protocol: UDP
Source Port: 22222
Destination Port: 44444
Found Fou over GRETAP
Key: 22222
GREfou tunnel 1 builded.
Byte code:
   0  3a ad a0 0d 89 e2 0e dd 91 c0 eb 09 0
  16  01 7a b6 a6 40 00 7e 11 2b e6 8c 72 0
```

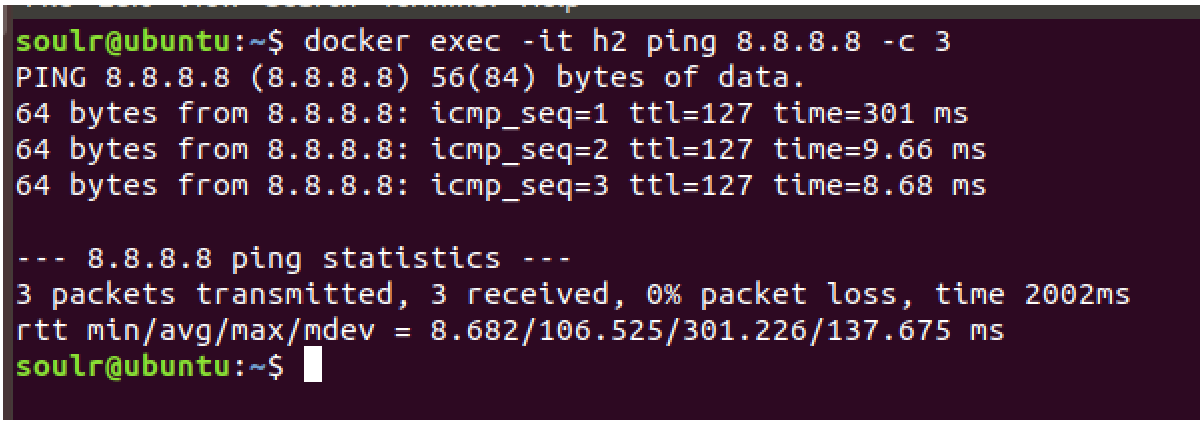
(b) Briefly explain how BRGr forwards packets back to the correct BRG via the corresponding GRETAP

After first time forward packet, it updates ARP table, then next time BRGr knows where it needs to send the packet to.



```
soulr@ubuntu:~/Desktop/project1$ docker exec -it BRGr brctl showmacs br0
port no mac addr                is local?       ageing timer
  1     06:08:26:ce:f8:6b       no                 7.61
  3     3e:f5:f4:4e:49:c9       no                 8.89
  2     7e:20:20:8c:29:a6       yes                0.00
  2     7e:20:20:8c:29:a6       yes                0.00
  2     a6:c2:71:a8:8d:e2       no                12.47
  1     e2:03:b0:6c:55:b3       yes                0.00
  1     e2:03:b0:6c:55:b3       yes                0.00
  3     e6:24:ec:d3:96:cf       yes                0.00
  3     e6:24:ec:d3:96:cf       yes                0.00
soulr@ubuntu:~/Desktop/project1$ docker exec -it h1 ifconfig h1BRg1veth
h1BRg1veth: error fetching interface information: Device not found
soulr@ubuntu:~/Desktop/project1$ docker exec -it h1 ifconfig h1BRG1veth
h1BRG1veth Link encap:Ethernet  HWaddr a6:c2:71:a8:8d:e2
          inet addr:20.0.1.2  Bcast:20.0.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1946 (1.9 KB)  TX bytes:1172 (1.1 KB)

soulr@ubuntu:~/Desktop/project1$ docker exec -it h2 ifconfig h2BRG2veth
h2BRG2veth Link encap:Ethernet  HWaddr 3e:f5:f4:4e:49:c9
          inet addr:20.0.1.3  Bcast:20.0.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1286 (1.2 KB)  TX bytes:1172 (1.1 KB)

soulr@ubuntu:~/Desktop/project1$
```

(c) Show that H2 can ping google 8.8.8.8

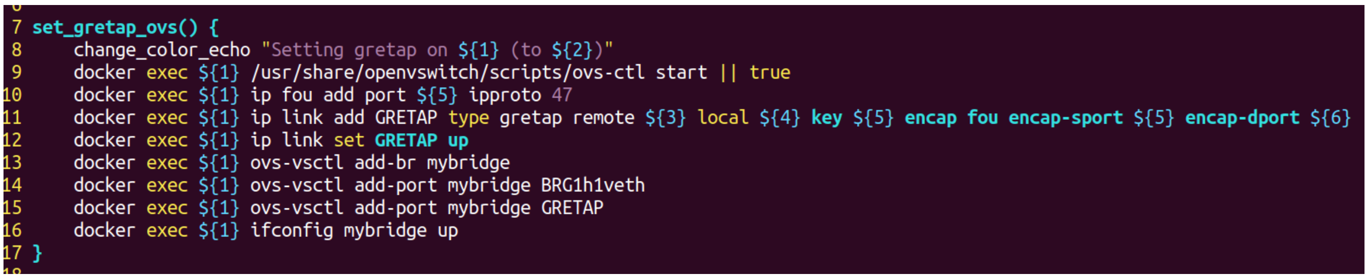


```
soulr@ubuntu:~$ docker exec -it h2 ping 8.8.8.8 -c 3
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=301 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=9.66 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=127 time=8.68 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 8.682/106.525/301.226/137.675 ms
soulr@ubuntu:~$
```

Part 2

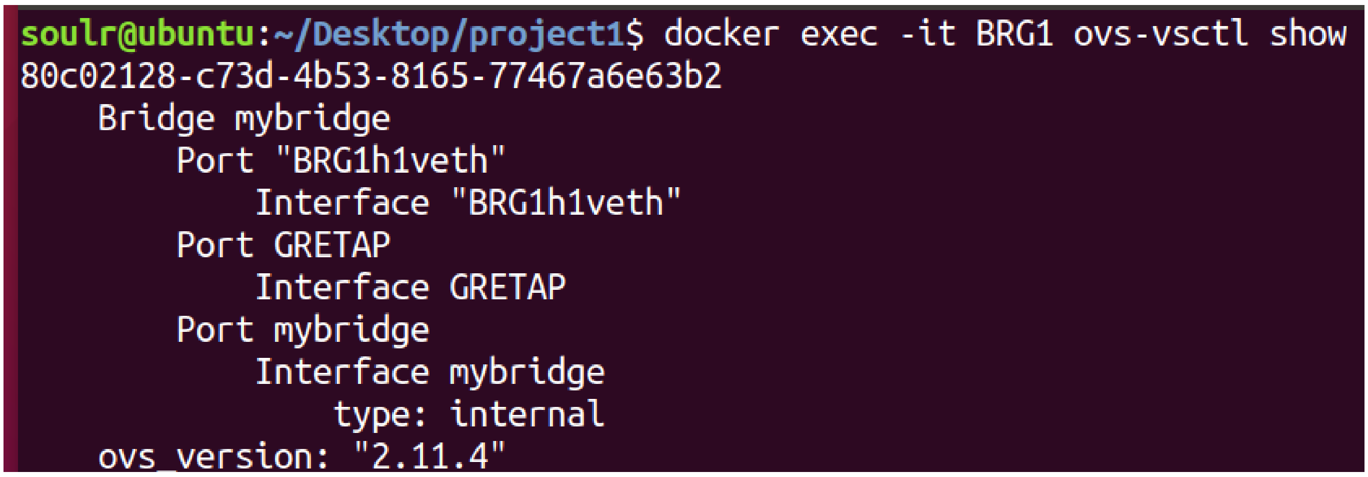
1. GRETAP on OVS bridge (5%)

(a) Show the configuration commands



```
set_gretap_ovs() {
    change_color_echo "Setting gretap on ${1} (to ${2})"
    docker exec ${1} /usr/share/openvswitch/scripts/ovs-ctl start || true
    docker exec ${1} ip fou add port ${5} ipproto 47
    docker exec ${1} ip link add GRETAP type gretap remote ${3} local ${4} key ${5} encap fou encap-sport ${5} encap-dport ${6}
    docker exec ${1} ip link set GRETAP up
    docker exec ${1} ovs-vsctl add-br mybridge
    docker exec ${1} ovs-vsctl add-port mybridge BRG1h1veth
    docker exec ${1} ovs-vsctl add-port mybridge GRETAP
    docker exec ${1} ifconfig mybridge up
}
```

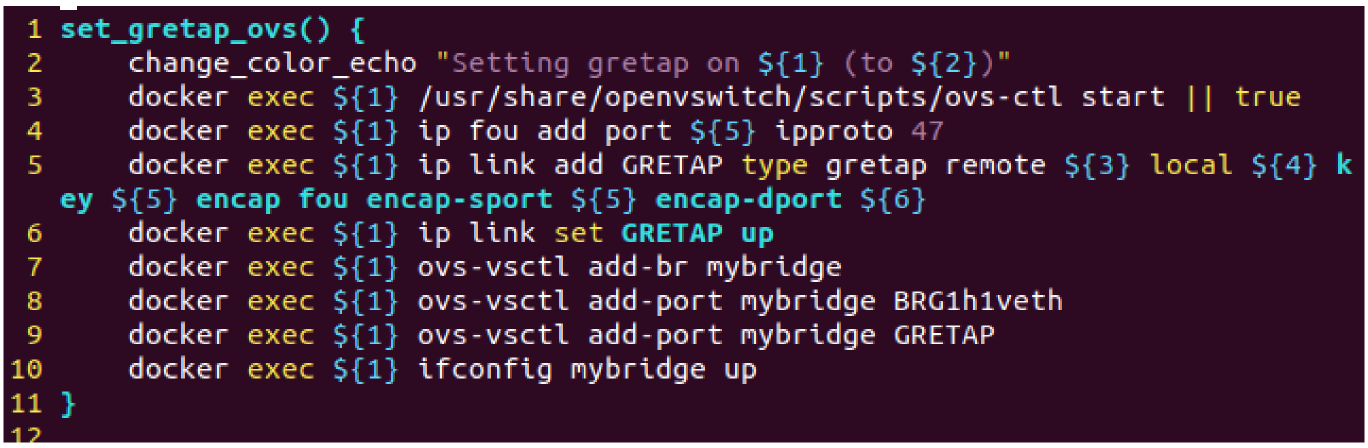
(b) Show the interfaces on OVS bridge



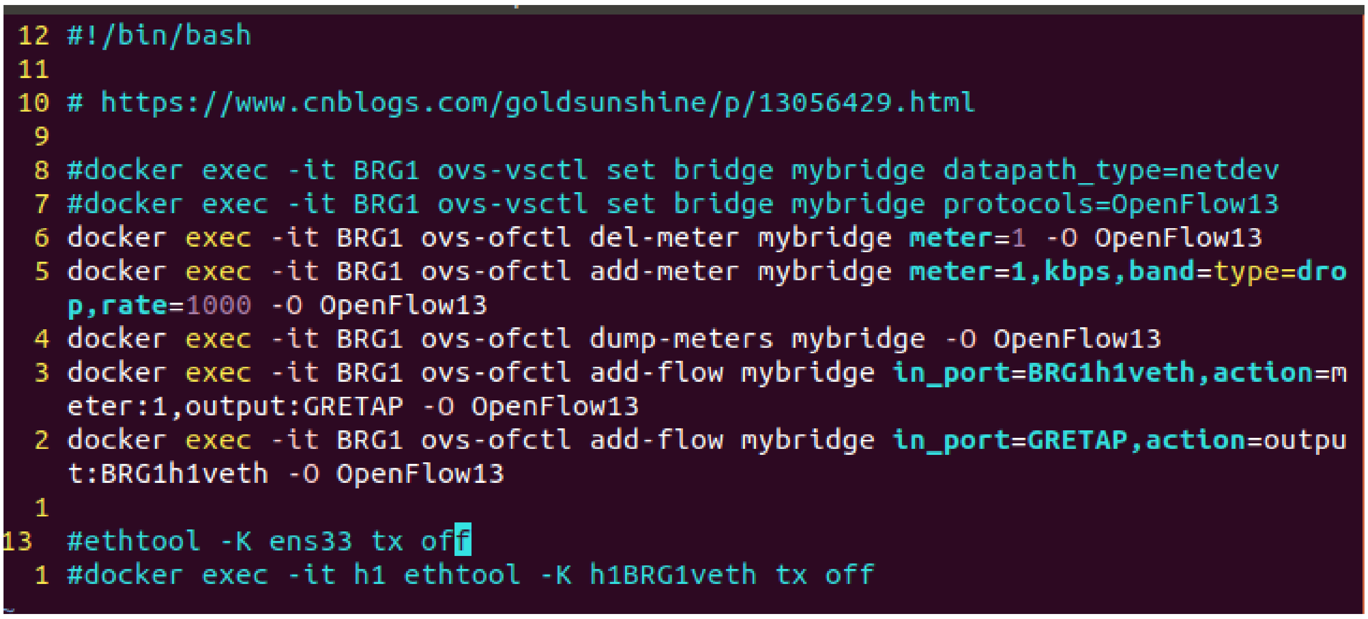
```
soulr@ubuntu:~/Desktop/project1$ docker exec -it BRG1 ovs-vsctl show
80c02128-c73d-4b53-8165-77467a6e63b2
    Bridge mybridge
        Port "BRG1h1veth"
            Interface "BRG1h1veth"
        Port GRETAP
            Interface GRETAP
        Port mybridge
            Interface mybridge
                type: internal
    ovs_version: "2.11.4"
```

2. Meter table (5%)

(a) Show the configuration commands

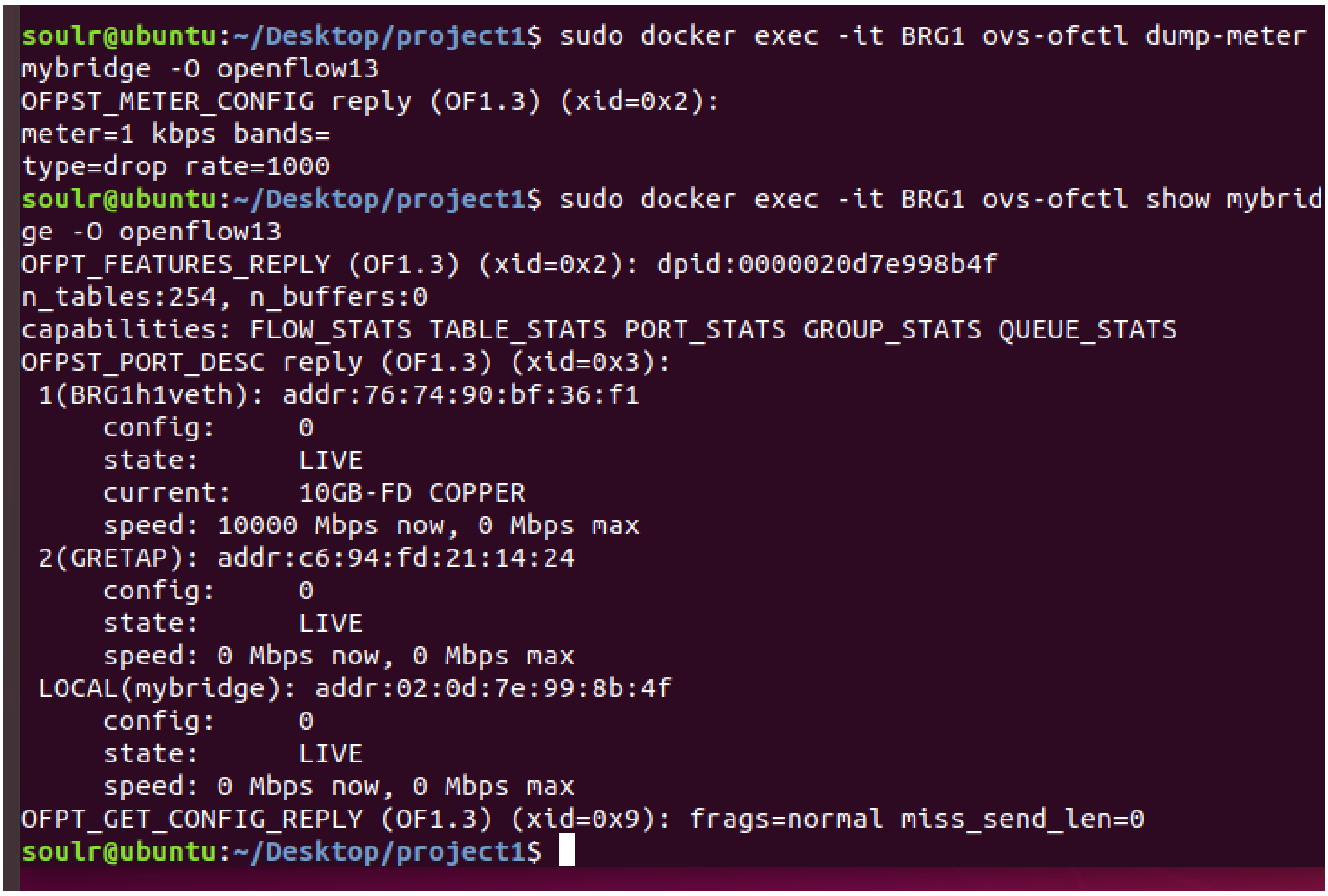


```
set_gretap_ovs() {
    change_color_echo "Setting gretap on ${1} (to ${2})"
    docker exec ${1} /usr/share/openvswitch/scripts/ovs-ctl start || true
    docker exec ${1} ip fou add port ${5} ipproto 47
    docker exec ${1} ip link add GRETAP type gretap remote ${3} local ${4} key ${5} encap fou encap-sport ${5} encap-dport ${6}
    docker exec ${1} ip link set GRETAP up
    docker exec ${1} ovs-vsctl add-br mybridge
    docker exec ${1} ovs-vsctl add-port mybridge BRG1h1veth
    docker exec ${1} ovs-vsctl add-port mybridge GRETAP
    docker exec ${1} ifconfig mybridge up
}
```



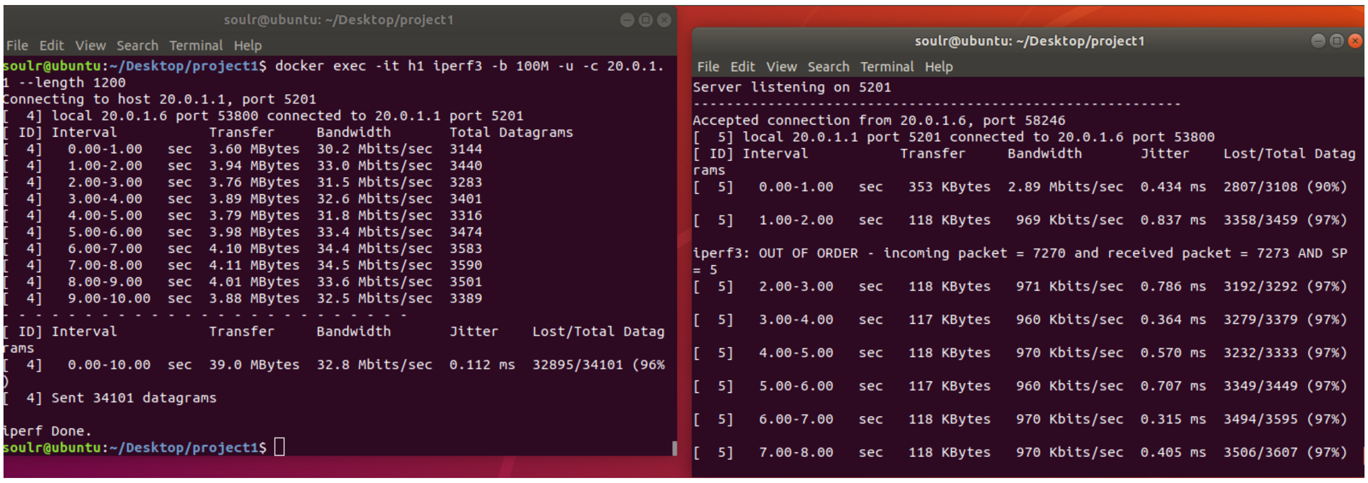
```
#!/bin/bash

# https://www.cnblogs.com/goldsunshine/p/13056429.html

#docker exec -it BRG1 ovs-vsctl set bridge mybridge datapath_type=netdev
#docker exec -it BRG1 ovs-vsctl set bridge mybridge protocols=OpenFlow13
docker exec -it BRG1 ovs-ofctl del-meter mybridge meter=1 -O OpenFlow13
docker exec -it BRG1 ovs-ofctl add-meter mybridge meter=1,kbps,band=type=drop,rate=1000 -O OpenFlow13
docker exec -it BRG1 ovs-ofctl dump-meters mybridge -O OpenFlow13
docker exec -it BRG1 ovs-ofctl add-flow mybridge in_port=BRG1h1veth,action=meter:1,output:GRETAP -O OpenFlow13
docker exec -it BRG1 ovs-ofctl add-flow mybridge in_port=GRETAP,action=output:BRG1h1veth -O OpenFlow13

#ethtool -K ens33 tx off
#docker exec -it h1 ethtool -K h1BRG1veth tx off
```

(b) Show meter entries and flow entries



```
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it BRG1 ovs-ofctl dump-meter
mybridge -O openflow13
OFPST_METER_CONFIG reply (OF1.3) (xid=0x2):
meter=1 kbps bands=
type=drop rate=1000
soulr@ubuntu:~/Desktop/project1$ sudo docker exec -it BRG1 ovs-ofctl show mybrid
ge -O openflow13
OFPT_FEATURES_REPLY (OF1.3) (xid=0x2): dpid:0000020d7e998b4f
n_tables:254, n_buffers:0
capabilities: FLOW_STATS TABLE_STATS PORT_STATS GROUP_STATS QUEUE_STATS
OFPST_PORT_DESC reply (OF1.3) (xid=0x3):
 1(BRG1h1veth): addr:76:74:90:bf:36:f1
     config:     0
     state:      LIVE
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 2(GRETAP): addr:c6:94:fd:21:14:24
     config:     0
     state:      LIVE
     speed: 0 Mbps now, 0 Mbps max
 LOCAL(mybridge): addr:02:0d:7e:99:8b:4f
     config:     0
     state:      LIVE
     speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (OF1.3) (xid=0x9): frags=normal miss_send_len=0
soulr@ubuntu:~/Desktop/project1$
```

(c) Observe and record the traffic speeds



```
soulr@ubuntu:~/Desktop/project1$ docker exec -it h1 iperf3 -b 100M -u -c 20.0.1.
1 --length 1200
Connecting to host 20.0.1.1, port 5201
[  4] local 20.0.1.6 port 53800 connected to 20.0.1.1 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.00   sec  3.60 MBytes  30.2 Mbits/sec  3144
[  4]   1.00-2.00   sec  3.94 MBytes  33.0 Mbits/sec  3440
[  4]   2.00-3.00   sec  3.76 MBytes  31.5 Mbits/sec  3283
[  4]   3.00-4.00   sec  3.89 MBytes  32.6 Mbits/sec  3401
[  4]   4.00-5.00   sec  3.79 MBytes  31.8 Mbits/sec  3316
[  4]   5.00-6.00   sec  3.98 MBytes  33.4 Mbits/sec  3474
[  4]   6.00-7.00   sec  4.10 MBytes  34.4 Mbits/sec  3583
[  4]   7.00-8.00   sec  4.11 MBytes  34.5 Mbits/sec  3590
[  4]   8.00-9.00   sec  4.01 MBytes  33.6 Mbits/sec  3501
[  4]   9.00-10.00  sec  3.88 MBytes  32.5 Mbits/sec  3389
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
[  4]   0.00-10.00  sec  39.0 MBytes  32.8 Mbits/sec  0.112 ms  32895/34101 (96%
)
[  4] Sent 34101 datagrams

iperf Done.
soulr@ubuntu:~/Desktop/project1$
```

```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 20.0.1.6, port 58246
[  5] local 20.0.1.1 port 5201 connected to 20.0.1.6 port 53800
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
[  5]   0.00-1.00   sec   353 KBytes  2.89 Mbits/sec  0.434 ms  2807/3108 (90%)

[  5]   1.00-2.00   sec   118 KBytes   969 Kbits/sec  0.837 ms  3358/3459 (97%)

iperf3: OUT OF ORDER - incoming packet = 7270 and received packet = 7273 AND SP
= 5
[  5]   2.00-3.00   sec   118 KBytes   971 Kbits/sec  0.786 ms  3192/3292 (97%)

[  5]   3.00-4.00   sec   117 KBytes   960 Kbits/sec  0.364 ms  3279/3379 (97%)

[  5]   4.00-5.00   sec   118 KBytes   970 Kbits/sec  0.570 ms  3232/3333 (97%)

[  5]   5.00-6.00   sec   117 KBytes   960 Kbits/sec  0.707 ms  3349/3449 (97%)

[  5]   6.00-7.00   sec   118 KBytes   970 Kbits/sec  0.315 ms  3494/3595 (97%)

[  5]   7.00-8.00   sec   118 KBytes   970 Kbits/sec  0.405 ms  3506/3607 (97%)
```