

1. Show the ping results to test reachability (5%)

h1 and h2 ping GWr

```
mininet> h1 ping GWr -c 1
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.110 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.110/0.110/0.110/0.000 ms
mininet> h2 ping GWr -c 1
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.091 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.091/0.091/0.091/0.000 ms
```

2. Show all interfaces of Node BRGr after h1 and h2 can ping GWr(5%)

Before	After
<pre>0 Name: BRGr-eth0 1 Name: br0 2 Name: BRGr-eth1 3 Name: any 4 Name: lo 5 Name: nflog 6 Name: nfqueue 7 Name: usbmon1 8 Name: usbmon2</pre>	<pre>0 Name: BRGr-eth0 1 Name: br0 2 Name: BRGr-eth1 3 Name: GRE_h1 4 Name: GRE_h2 5 Name: any 6 Name: lo 7 Name: nflog 8 Name: nfqueue 9 Name: usbmon1 10 Name: usbmon2</pre>

3. Draw the interconnection diagram of interfaces and Linux bridge on BRGr. Explain your diagram with the screenshot of interface list of BRGr. (10%)

diagram	screenshot
<p>BRGr interconnection diagram</p>	<pre>0 Name: BRGr-eth0 1 Name: br0 2 Name: BRGr-eth1 3 Name: GRE_h1 4 Name: GRE_h2 5 Name: any 6 Name: lo 7 Name: nflog 8 Name: nfqueue 9 Name: usbmon1 10 Name: usbmon2</pre>
Explain	
<p>Add a bridge br0 and connect it to BRGr-eth0.</p> <p>After receiving GRE packet , set up gretap interfaces GRE_h1 for 140.114.0.1 and GRE_h2 for 140.115.0.1.</p>	

4. Explain how Linux kernel of BRGr determines which gretap interface to forward packets from GWr to hosts (h1 or h2)?

Describe your answer with appropriate screenshot. (10%)

screenshot	
<pre>root@SDN-NFV:~/mininet# brctl showmacs br0 port no mac addr is local? ageing timer 1 02:39:43:5f:82:38 yes 0.00 1 02:39:43:5f:82:38 yes 0.00 3 82:3c:3b:a1:0e:fe yes 0.00 3 82:3c:3b:a1:0e:fe yes 0.00 2 b2:44:cb:13:13:65 yes 0.00 2 b2:44:cb:13:13:65 yes 0.00 root@SDN-NFV:~/mininet# ip link show GRE_h1 9: GRE_h1@NONE: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1462 q r br0 state UNKNOWN mode DEFAULT group default qlen 1000 link/ether 82:3c:3b:a1:0e:fe brd ff:ff:ff:ff:ff:ff root@SDN-NFV:~/mininet# ip link show GRE_h2 8: GRE_h2@NONE: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1462 q r br0 state UNKNOWN mode DEFAULT group default qlen 1000 link/ether b2:44:cb:13:13:65 brd ff:ff:ff:ff:ff:ff</pre>	
Explain	
<p>Linux bridge learns MAC address like bridge and switch.</p> <p>The MAC address for a frame is learned when the frame enters the bridge through an interface. The MAC address is recorded in the bridge table and looked up when the bridge forwarding frames.</p>	
Other details	
<p>The interfaces with 'is local' as Yes are interfaces directly on the Linux Bridge including tap interfaces.</p> <p>In the screenshot, Port 1 is connected to eth0; Port 2 represent the tap interfaces GRE_h2, and Port 3 represent GRE_h1 on br0.</p>	

5. Run tcpdump on h1 to capture packet and take screenshot to explain why or why not h1 is aware of GRE tunneling. (10%)

screenshot	
<pre>root@SDN-NFV:~/mininet# tcpdump icmp -n tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on h1-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes 15:35:49.010533 IP 10.0.0.1 > 10.0.0.3: ICMP echo request, id 4129, seq 1, length 64 15:35:49.010601 IP 10.0.0.3 > 10.0.0.1: ICMP echo reply, id 4129, seq 1, length 64</pre>	
Explain	
<p>The payload packet with private IP is encapsulated with GRE and delivery header by BRG1; then forward to BRGr.</p> <p>BRGr extracts the payload packet and routes it to GWr with private IP afterwards.</p> <p>Since the tunnel is implemented by routers, the endpoint hosts don't need to be aware of the tunnel.</p>	