

網路系統總整與實作 Lab #4

GRE Tunnel and Auto Creation

0716236 劉耀文

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

a) 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=0.131 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.131/0.131/0.131/0.000 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=0.112 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.112/0.112/0.112/0.000 ms
```

➔ 此圖為 BRGr 自動建立 GRE Tunnel 後，h1 以及 h2 透過 ping 指令向 GWr 溝通，並成功收到 ICMP reply 封包的截圖。由於建立好 GRE Tunnel，所以 h1、GWr 以及 h2、GWr 可以分別視為一個子網域，因此可以看到在執行 ping 指令時的 IP 為子網段 IP，也可以透過外部網路進行溝通。

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

```
mininet> BRGr ifconfig
BRGr-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.8 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::a42b:81ff:feef:6bb0 prefixlen 64 scopeid 0x20<link>
    ether a6:2b:81:ef:6b:b0 txqueuelen 1000 (Ethernet)
    RX packets 18 bytes 1340 (1.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 35 bytes 2634 (2.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

BRGr-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 140.113.0.1 netmask 255.255.0.0 broadcast 140.113.255.255
    inet6 fe80::c8bf:32ff:fed0:ac8b prefixlen 64 scopeid 0x20<link>
    ether ca:bf:32:d0:ac:8b txqueuelen 1000 (Ethernet)
    RX packets 65 bytes 6660 (6.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 102 bytes 11540 (11.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

GRE1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1462
    inet6 fe80::b085:c7ff:feed:6950 prefixlen 64 scopeid 0x20<link>
    ether b2:85:c7:ed:69:50 txqueuelen 1000 (Ethernet)
    RX packets 8 bytes 532 (532.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 1752 (1.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

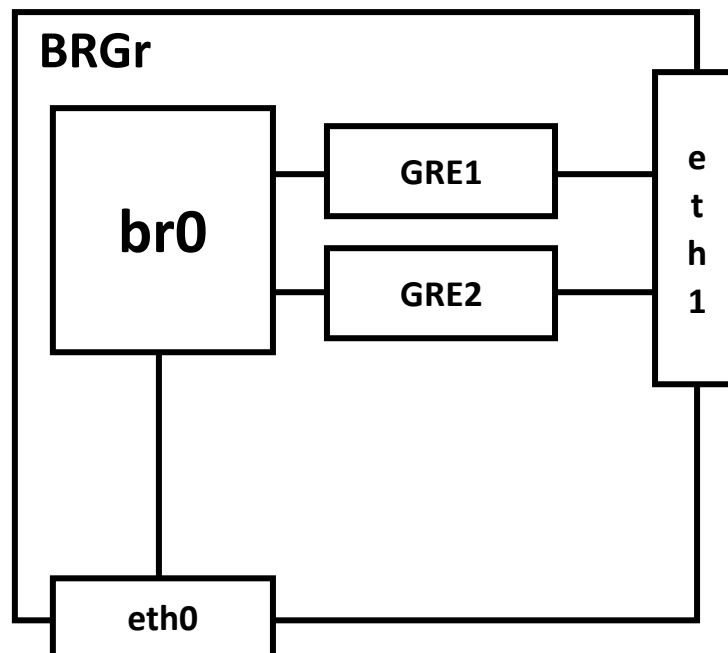
GRE2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1462
    inet6 fe80::5042:99ff:fe06:6576 prefixlen 64 scopeid 0x20<link>
    ether 52:42:99:06:65:76 txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 392 (392.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 31 bytes 1920 (1.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1462
    inet6 fe80::5042:99ff:fe06:6576 prefixlen 64 scopeid 0x20<link>
    ether 52:42:99:06:65:76 txqueuelen 1000 (Ethernet)
    RX packets 11 bytes 560 (560.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 914 (914.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

➔ 此圖為 h1 以及 h2 可以分別 ping 到 GWr 之後，BRGr 的 network interface list。其中 BRGr-eth0 以及 BRGr-eth1 為 topology.py 所定義的裝置，在 auto creation 的程式中不會修改此二介面。GRE1 以及 GRE2 則是透過 auto creation 程式過濾 GRE 封包後建立而成，此二介面分別和 BRG1 以及 BRG2 建立 GRE Tunnel，所有來自此二 host 的封包皆會由此進入。br0 則是將 GRE1 以及 GRE2 和 BRGr-eth0 間做為橋接用途之裝置，若無此裝置則在 GRE Tunnel 建立後無法順利傳輸。最後 lo 則為 loopback interface，此一介面為 localhost 所使用，在這次 Lab 當中不會使用。

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



➔ 此圖為 BRGr 內各網路介面間的連接圖。由上題可以得知，eth0 以及 eth1 為對外之溝通介面，GRE1 以及 GRE2 分別為對 BRG1 以及 BRG2 的 GRE Tunnel interface，主要用來拆開或是封裝 GRE 封包，用來達到 gretap 的效果，將 h1、GWr 以及 h2、GWr 分別視為處於同一個子網域。然而 GRE interface 仍需 br0 這個 Linux 內建之橋接器，將之與 eth0 連接後整個 GRE Tunnel 方能使用。

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%)

```

root@SDN-NFV: ~/NSCap/Lab/Lab4
root@SDN-NFV:~/NSCap/Lab/Lab4# brctl showmacs br0
port no mac addr is local? ageing timer
3 36:04:a1:e5:13:56 no 31.96
2 42:84:73:78:03:d8 no 42.20
1 46:c9:e8:65:e6:a5 no 31.96
3 52:42:99:06:65:76 yes 0.00
3 52:42:99:06:65:76 yes 0.00
2 76:4f:22:80:d0:06 no 13.54
2 8e:4e:cd:37:a4:e5 no 11.48
1 a6:2b:81:ef:6b:b0 yes 0.00
1 a6:2b:81:ef:6b:b0 yes 0.00
3 ae:a7:22:8f:2f:0f no 36.07
2 b2:85:c7:ed:69:50 yes 0.00
2 b2:85:c7:ed:69:50 yes 0.00
root@SDN-NFV:~/NSCap/Lab/Lab4#

```

<pre> Ethernet type: IPv4 Outer Src IP: 140.114.0.1 Outer Dst IP: 140.113.0.1 Next Layer Protocol: GRE Protocol type: 0x6558 Inner Source MAC: 42:84:73:78:03:D8 Inner Destination MAC: 33:33:00:00:00:02 </pre>	<pre> Ethernet type: IPv4 Outer Src IP: 140.115.0.1 Outer Dst IP: 140.113.0.1 Next Layer Protocol: GRE Protocol type: 0x6558 Inner Source MAC: ae:a7:22:8f:2f:0f Inner Destination MAC: 33:33:00:00:00:02 </pre>
--	--

➔ 上圖為 BRGr 執行 brctl showmacs br0 指令後得到之 MAC Learning Address Table、下二圖為 auto creation 程式擷取之封包得到的結果。由上三圖可以得知 h1-eth0 以及 h2-eth0 的 MAC address 皆有出現在 table 上，也對應著不同的 port(這邊的 port2 為經過 GRE1、port3 為經過 GRE2)，由此可知 kernel 是透過 MAC learning 決定封包的路徑。

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

```
root@SDN-NFV: ~/NSCap/Lab/Lab4
root@SDN-NFV:~/NSCap/Lab/Lab4# tcpdump -i h1-eth0 icmp -XX -n -c 2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h1-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
16:39:40.916742 IP 10.0.0.1 > 10.0.0.3: ICMP echo request, id 2394, seq 1, length 64
0x0000: 1e74 1172 60c9 86f1 02fe 144e 0800 4500 .t.r`.....N..E.
0x0010: 0054 026b 4000 4001 243b 0a00 0001 0a00 .T.k@.@.$;.....
0x0020: 0003 0800 f004 095a 0001 cc6f 6d60 0000 .....Z...om ..
0x0030: 0000 f8fc 0d00 0000 0000 1011 1213 1415 .....
0x0040: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 .....! "$%
0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 &'()*+,-./012345
0x0060: 3637 67
16:39:40.916786 IP 10.0.0.3 > 10.0.0.1: ICMP echo reply, id 2394, seq 1, length 64
0x0000: 86f1 02fe 144e 1e74 1172 60c9 0800 4500 .....N.t.r`...E.
0x0010: 0054 34db 0000 4001 31cb 0a00 0003 0a00 .T4...@.1.....
0x0020: 0001 0000 f804 095a 0001 cc6f 6d60 0000 .....Z...om ..
0x0030: 0000 f8fc 0d00 0000 0000 1011 1213 1415 .....
0x0040: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 .....! "$%
0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 &'()*+,-./012345
0x0060: 3637 67
2 packets captured
2 packets received by filter
0 packets dropped by kernel
root@SDN-NFV:~/NSCap/Lab/Lab4#
```

- h1 並不知道 GRE Tunnel 的存在
- 此圖為執行 h1 ping GWr -c 1 後在 h1 以 tcpdump 擷取之兩個 icmp 封包並將之以 16 進位顯示於 CLI 介面當中之截圖。其中以藍色框起來的部分為 Ethernet header、綠色框起來的部分為 IP header。我們可以從 Ethernet header 的最後兩個 byte 分辨下一層 header 的種類為何，由第一個紅色方框可以知道下一個 header 之種類為 IP header(0x0800)。接著在 IP header 當中，我們可以透過第 10 個 byte 來得知接下來的 header 為哪一種 header，由第二個紅色方框可以知道下一個 header 之種類為 ICMP header(0x01 (1)₁₀)，而非 GRE protocol header(0x2F (47)₁₀)，故我們可以推論出 h1 並不知道 GRE Tunnel 的存在，因為 GRE Tunnel 只作用於 BRG1 以及 BRGr 之間，在這之間傳輸所需用到之 GRE header 也已經被 BRG1 的 gretap interface 拔除。