

Lab 4: Overlay Network and VXLAN

孙济宸 520030910016

VXLAN setup

On VM 1: In mininet, set the IPs for both hosts:

```
h1 ifconfig h1-eth0 10.0.0.1 netmask 255.0.0.0
h2 ifconfig h2-eth0 10.0.0.2 netmask 255.0.0.0
```

Open another terminal, set the switch IP:

```
sudo ifconfig s1 10.0.0.5/8 up
```

On VM 2:

In mininet, set the IPs for both hosts:

```
h3 ifconfig h3-eth0 10.0.0.3 netmask 255.0.0.0
h4 ifconfig h4-eth0 10.0.0.4 netmask 255.0.0.0
```

Open another terminal, set the switch IP:

```
sudo ifconfig s2 10.0.0.6/8 up
```

When we ping 10.0.0.6/10.0.0.3/10.0.0.4 from 10.0.0.5/10.0.0.1/10.0.0.2, it returns 'unreachable'.

On VM1:

```
sudo ovs-vsctl add-br br1
```

Assign the IP of ens34 to br1 :

```
sudo ovs-vsctl add-port br1 ens34
sudo ifconfig br1 192.168.56.101/24 up
```

Add a new default route for br1:

```
sudo route add default gw 192.168.56.201
```

On VM2:

```
sudo ovs-vsctl add-br br1
```

Assign the IP of ens34 to br1:

```
sudo ovs-vsctl add-port br1 ens34  
sudo ifconfig br1 192.168.56.103/24 up
```

Add a new default route for br1:

```
sudo route add default gw 192.168.56.201
```

Now we create the VXLAN tunnel.

On VM1:

```
sudo ovs-vsctl add-port s1 vxlan1 -- set interface vxlan1 type=vxlan  
options:remote_ip=192.168.56.103
```

On VM2:

```
sudo ovs-vsctl add-port s2 vxlan1 -- set interface vxlan1 type=vxlan  
options:remote_ip=192.168.56.101
```

Testing

After this we we can test ping and the network works.

```
madcreeper@madcreeper-VirtualBox: ~/lab4
File Edit View Search Terminal Help
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=13.5 ms
^C
--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 13.571/14.148/15.262/0.787 ms
mininet> h1 ping 10.0.0.6
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_seq=1 ttl=64 time=35.3 ms
64 bytes from 10.0.0.6: icmp_seq=2 ttl=64 time=12.5 ms
64 bytes from 10.0.0.6: icmp_seq=3 ttl=64 time=11.5 ms
64 bytes from 10.0.0.6: icmp_seq=4 ttl=64 time=11.6 ms
^C
--- 10.0.0.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 11.595/17.793/35.332/10.134 ms
mininet> h1 ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=64.9 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=25.3 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=24.3 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=38.9 ms
^C
--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 24.387/38.401/64.965/16.379 ms
mininet> h1 ping 10.0.0.3
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=54.3 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=25.3 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=24.9 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=29.2 ms
^C
--- 10.0.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 24.960/33.483/54.395/12.191 ms
mininet>
```

Using *Wireshark*, we can see that the ping command uses these protocols below:

The image shows a Wireshark packet capture on interface eth0. The packet list pane displays several ICMP Echo (ping) requests and replies, as well as ARP requests. The packet details pane shows the structure of the first packet, which is an ICMP Echo (ping) request. The packet bytes pane shows the raw data of the first packet, which is an ICMP Echo (ping) request.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.5	10.0.0.6	ICMP	98	Echo (ping) request id=0x0000, seq=1/256, ttl=64 (request in 2)
2	0.000194966	10.0.0.6	10.0.0.5	ICMP	98	Echo (ping) reply id=0x0000, seq=1/256, ttl=64 (request in 1)
3	1.002281274	10.0.0.5	10.0.0.6	ICMP	98	Echo (ping) request id=0x0000, seq=2/512, ttl=64 (reply in 4)
4	1.005437528	10.0.0.6	10.0.0.5	ICMP	98	Echo (ping) reply id=0x0000, seq=2/512, ttl=64 (request in 3)
5	2.010268267	10.0.0.5	10.0.0.6	ICMP	98	Echo (ping) request id=0x0000, seq=3/768, ttl=64 (reply in 6)
6	2.011735708	10.0.0.6	10.0.0.5	ICMP	98	Echo (ping) reply id=0x0000, seq=3/768, ttl=64 (request in 5)
7	3.011320876	10.0.0.5	10.0.0.6	ICMP	98	Echo (ping) request id=0x0000, seq=4/1024, ttl=64 (reply in 8)
8	3.011947698	10.0.0.6	10.0.0.5	ICMP	98	Echo (ping) reply id=0x0000, seq=4/1024, ttl=64 (request in 7)
9	5.139964061	ce:9e:49:54:b9:43	9e:8f:63:ab:04:42	ARP	42	Who has 10.0.0.6? Tell 10.0.0.5
10	5.148511419	9e:8f:63:ab:04:42	ce:9e:49:54:b9:43	ARP	42	10.0.0.6 is at 9e:8f:63:ab:04:42
11	5.148607948	9e:8f:63:ab:04:42	ce:9e:49:54:b9:43	ARP	42	Who has 10.0.0.5? Tell 10.0.0.6
12	5.148614583	ce:9e:49:54:b9:43	9e:8f:63:ab:04:42	ARP	42	10.0.0.5 is at ce:9e:49:54:b9:43

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0
Ethernet II, Src: ce:9e:49:54:b9:43 (ce:9e:49:54:b9:43), Dst: 9e:8f:63:ab:04:42 (9e:8f:63:ab:04:42)
Internet Protocol Version 4, Src: 10.0.0.5, Dst: 10.0.0.6
Internet Control Message Protocol

0000 9e 8f 63 ab 04 42 ce 9e 49 54 b9 43 08 00 45 00 --c--B--ITC-E
0010 00 54 40 92 40 00 01 e6 0c 0a 00 00 05 0a 00 --T0@0-----
0020 00 06 08 00 b7 38 0e ed 00 01 f7 60 93 62 00 00 ---8-...b-
0030 00 00 e1 42 07 00 00 00 00 10 11 12 13 14 15 --B-----
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25!##\$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345
0060 36 37 67

The screenshot shows a Wireshark capture on interface `enp0s8`. The packet list displays 16 packets, including ICMP echo requests and replies between `10.0.0.5` and `10.0.0.6`, and ARP requests from `10.0.0.5` to `10.0.0.6`. The packet details pane for packet 16 shows the structure of a VXLAN packet, including the Ethernet II header, the IPv4 header, and the VXLAN header.

- ICMP: Mainly used by the ping command, namely ICMP echo request and reply.
- IPv4. The network layer protocol used in this lab.
- ARP. This is used to resolve the MAC address given an IP address. We can see in the screenshot that a host (s1) asks for the MAC address of 10.0.0.6 and gets a response.
- Ethernet: the data link layer protocol used by Mininet.

Before we use *iperf* to test the bandwidth, we need to change the MTU of the hosts:

```
mininet> h1 ifconfig h1-eth0 mtu 1450
mininet> h2 ifconfig h2-eth0 mtu 1450
```

```
^Cmininet> h4 ifconfig h4-eth0 mtu 1450
```

This is because VXLAN adds a 50 Byte header to a data link layer segment. In order for *iperf* to work on a standard mtu of 1500, we need to reduce the host ethernet interface mtu.

The bandwidth between s1 and s2 is an unlimited 1Gbps.

```
madcreeper@madcreeper-VirtualBox:~$ iperf3 -c 192.168.56.103 -p 5201
Connecting to host 192.168.56.103, port 5201
[ 4] local 192.168.56.101 port 45450 connected to 192.168.56.103 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr   Cwnd
[ 4]  0.00-1.00      sec    129 MBytes    1.08 Gbits/sec    394    249 KBytes
[ 4]  1.00-2.00      sec    138 MBytes    1.16 Gbits/sec    351    235 KBytes
[ 4]  2.00-3.00      sec    141 MBytes    1.19 Gbits/sec    300    246 KBytes
[ 4]  3.00-4.00      sec    138 MBytes    1.16 Gbits/sec    264    199 KBytes
[ 4]  4.00-5.00      sec    141 MBytes    1.18 Gbits/sec    224    252 KBytes
[ 4]  5.00-6.00      sec    138 MBytes    1.16 Gbits/sec    224    249 KBytes
[ 4]  6.00-7.00      sec    141 MBytes    1.18 Gbits/sec    289    238 KBytes
[ 4]  7.00-8.00      sec    128 MBytes    1.07 Gbits/sec    247    184 KBytes
[ 4]  8.00-9.00      sec    134 MBytes    1.13 Gbits/sec    242    214 KBytes
[ 4]  9.00-10.00     sec    134 MBytes    1.13 Gbits/sec    217    242 KBytes
- - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[ 4]  0.00-10.00     sec    1.33 GBytes    1.14 Gbits/sec    2752
[ 4]  0.00-10.00     sec    1.33 GBytes    1.14 Gbits/sec
sender
receiver

iperf Done.
madcreeper@madcreeper-VirtualBox:~$
```

The bandwidth between h1/h2 and h3/h4 is limited by the 10Mbps links.

```
madcreeper@madcreeper-VirtualBox: ~/lab4
File Edit View Search Terminal Help
4 packets transmitted, 4 received, 0% packet loss, time 301ms
rtt min/avg/max/mdev = 0.693/3.443/8.176/2.905 ms
mininet> h1 ifconfig h1-eth0 mtu 1450
mininet> h2 ifconfig h2-eth0 mtu 1450
mininet> s1 ifconfig s1-eth0 mtu 1450
SIOCSIFMTU: No such device
mininet> h1 iperf3 -c 10.0.0.4 -p 5201
Connecting to host 10.0.0.4, port 5201
[ 4] local 10.0.0.1 port 47110 connected to 10.0.0.4 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr   Cwnd
[ 4]  0.00-1.00      sec    1.48 MBytes    12.4 Mbits/sec     0    94.2 KBytes
[ 4]  1.00-2.00      sec    1.29 MBytes    10.8 Mbits/sec     0    145 KBytes
[ 4]  2.00-3.01      sec    1.29 MBytes    10.7 Mbits/sec     0    198 KBytes
[ 4]  3.01-4.01      sec    1.29 MBytes    10.8 Mbits/sec     0    248 KBytes
[ 4]  4.01-5.00      sec    1.29 MBytes    10.8 Mbits/sec     0    299 KBytes
[ 4]  5.00-6.01      sec    1.35 MBytes    11.2 Mbits/sec     0    352 KBytes
[ 4]  6.01-7.00      sec    1.23 MBytes    10.3 Mbits/sec     0    404 KBytes
[ 4]  7.00-8.01      sec    1.29 MBytes    10.8 Mbits/sec     0    455 KBytes
[ 4]  8.01-9.00      sec    1.29 MBytes    10.8 Mbits/sec     0    505 KBytes
[ 4]  9.00-10.01     sec    1.29 MBytes    10.7 Mbits/sec     0    556 KBytes
- - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[ 4]  0.00-10.01     sec    13.1 MBytes    11.0 Mbits/sec     0
[ 4]  0.00-10.01     sec    10.5 MBytes    8.80 Mbits/sec
sender
receiver

iperf Done.
mininet> h2 iperf3 -c 10.0.0.4 -p 5201
Connecting to host 10.0.0.4, port 5201
[ 4] local 10.0.0.2 port 51908 connected to 10.0.0.4 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr   Cwnd
[ 4]  0.00-1.01      sec    1.72 MBytes    14.3 Mbits/sec     0    126 KBytes
[ 4]  1.01-2.00      sec    1.35 MBytes    11.4 Mbits/sec     0    179 KBytes
[ 4]  2.00-3.01      sec    1.29 MBytes    10.8 Mbits/sec     0    232 KBytes
[ 4]  3.01-4.00      sec    1.17 MBytes    9.83 Mbits/sec     0    279 KBytes
[ 4]  4.00-5.00      sec    1.29 MBytes    10.8 Mbits/sec     0    329 KBytes
[ 4]  5.00-6.00      sec    1.35 MBytes    11.3 Mbits/sec     0    384 KBytes
[ 4]  6.00-7.00      sec    1.41 MBytes    11.8 Mbits/sec     0    438 KBytes
[ 4]  7.00-8.00      sec    1.35 MBytes    11.3 Mbits/sec     0    493 KBytes
[ 4]  8.00-9.00      sec    1.23 MBytes    10.2 Mbits/sec     0    541 KBytes
[ 4]  9.00-10.00     sec    1.23 MBytes    10.3 Mbits/sec     0    590 KBytes
- - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[ 4]  0.00-10.00     sec    13.4 MBytes    11.2 Mbits/sec     0
[ 4]  0.00-10.00     sec    10.6 MBytes    8.92 Mbits/sec
sender
receiver

iperf Done.
mininet>
```

ping tells a similar story. Since each hop is 5ms except for between s1 and s2, the rtt is around 20-30 ms between h1/h2 and h3/h4, and < 1ms between s1 and s2.

```
mininet> h1 ping -c 4 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=25.1 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=32.2 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=25.4 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=24.0 ms

--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3014ms
rtt min/avg/max/mdev = 24.062/26.732/32.280/3.247 ms
mininet> h2 ping -c 4 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=26.2 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=41.0 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=28.2 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=29.2 ms

--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 26.285/31.200/41.026/5.775 ms
mininet> s1 ping -c 4 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=31.5 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=13.5 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=11.2 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=12.9 ms

--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 11.283/17.345/31.546/8.242 ms
mininet> █
```

```
madcreeper@madcreeper-VirtualBox:~$ ping -c 4 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=1.57 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.647 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.624 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.597 ms

--- 192.168.56.103 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.597/0.860/1.573/0.412 ms
madcreeper@madcreeper-VirtualBox:~$ █
```