

## Lab 4: Overlay Network and VXLAN

### What is Overlay Network?

An overlay network can be thought of as a computer network on top of another network. All nodes in an overlay network are connected with one another by means of logical or virtual links and each of these links correspond to a path in the underlying network. You can refer to the following links for more details.

<https://www.digitaltut.com/vxlan-tutorial/2>

### Experiment with VXLAN

We need two virtual machines for this VXLAN experiment. In each VM, we use Mininet to set up a LAN with 1 host and 1 OVS switch. Note that both LANs in the two VMs are behind NAT. Hence, h1 cannot communicate with h2. The purpose of this lab is to use VXLAN to build a tunnel or an overlay network between these two LANs, so that h1-h2 can directly communicate.

First, we will use the two VMs created at lab1.

IP address on VM1:

enp0s8: 192.168.56.101.

IP address on VM2:

enp0s8: 192.168.56.102.

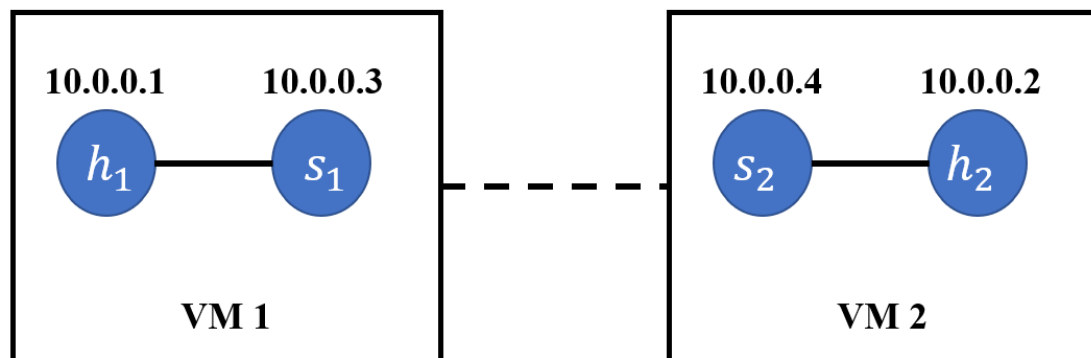


Figure 1

Please construct the network topology in Figure 1 using mininet on VM1 and VM2. (Set all the links bandwidth as 10Mbps, the packet loss rate as 0%, and the time delay as 5ms).

Set the IP addresses of hosts and switches as follows,

#### On VM 1:

In mininet, set the IPs for both hosts:

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

Open another terminal, set the switch IP:

```
sudo ifconfig s1 10.0.0.3/8 up
```

### **On VM 2:**

In mininet, set the IPs for both hosts:

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

Open another terminal, set the switch IP:

```
sudo ifconfig s2 10.0.0.4/8 up
```

Now, if you ping 10.0.0.2/10.0.0.4 from 10.0.0.1/10.0.0.3, there will be no response.

Then, we will use VXLAN to build a tunnel, through which h1-h2 can directly communicate.

### **Step 1:**

#### **On VM1:**

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

Assign the IP of enp0s8 to br1:

```
sudo ovs-vsctl add-port br1 enp0s8
```

```
sudo ifconfig br1 192.168.56.101/24 up
```

```
//Add a new default route for br1:
```

#### **On VM2:**

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

Assign the IP of enp0s8 to br1:

```
sudo ovs-vsctl add-port br1 enp0s8
```

```
sudo ifconfig br1 192.168.56.102/24 up
```

```
//Add a new default route for br1:
```

### **Step 2:**

#### **On VM1:**

```
//Build the VXLAN tunnel and set the remote IP address as 192.168.56.102:
```

#### **On VM2:**

```
//Build the VXLAN tunnel and set the remote IP address as 192.168.56.101:
```

Now, you can ping 10.0.0.2/10.0.0.4 from 10.0.0.1/10.0.0.3.

**Homework:**

1. (10 points) Add two lines of commands in Step 1 and two lines of commands in Step 2 to build the VXLAN tunnel between VM1 and VM2.
2. (20 points) Ping 10.0.0.2 from 10.0.0.1 (in your terminal of VM1) and use Wireshark to monitor the interfaces s1 and enp0s8, and describe the protocols used in this procedure, and how VXLAN works.

**Requirements:**

Please provide the screenshot of the wireshark interface in your report. Your description and findings should align with the information in wireshark interface.

3. (30 points) Use iperf3 to test the network bandwidth between the two virtual machines
  - Test the bandwidth between 10.0.0.3 and 10.0.0.4

Show the results and explain the reason.

**Requirements:**

Please provide the screenshot of your terminals in your report.

4. (40 points) Can you improve the bandwidth between 10.0.0.3 and 10.0.0.4, by the following two approaches?
  - 1). Designating the MTU size of iperf3.
  - 2). Designating the MTU of the two enp0s8 interfaces.

**Requirements:**

Please provide the screenshot of your terminals in your report.