

TẠO TOPOLOGY BẰNG MININET

I. MỤC ĐÍCH THÍ NGHIỆM

Bài thí nghiệm này giúp sinh viên làm quen với phần mềm Mininet và cách hoạt động của Open vSwitch. Qua bài thí nghiệm chúng ta tiến hành tạo một topology đơn giản bằng mininet và tạo các flow-entry cho Open vSwitch để các host có thể ping với nhau.

II. THẢO LUẬN

Mininet là một hệ thống cho phép thử nghiệm các mạng lớn trên một máy tính. Mininet hỗ trợ việc nghiên cứu, phát triển, kiểm thử, gỡ lỗi và các nhiệm vụ khác bằng việc giả lập một mạng các OpenFlow switch trên một máy tính duy nhất, nghĩa là ta có thể triển khai thử nghiệm mạng với hàng trăm/ngàn node chỉ trên một máy tính cá nhân. Trên mạng đó ta có thể triển khai các giao thức thử nghiệm trên mạng hoàn toàn giống như việc sử dụng các switch OpenFlow thật. Các OpenFlow switch mà mininet giả lập được gọi là Open vSwitch.

Một số chức năng của Mininet gồm có:

- Công cụ xây dựng môi trường giả lập của các OpenFlow switch, đơn giản và không tốn kém để phát triển các ứng dụng mạng. Do các OpenFlow switch trong Mininet có tất cả các tính chất mà OpenFlow switch thật có được nên việc sử dụng mạng giả lập bằng Mininet cho phép kiểm tra các chức năng của ứng dụng mà không cần thiết bị thật.
- Cho phép các nhà phát triển ứng dụng làm việc đồng thời, một cách độc lập trên cùng đồ hình mà không ảnh hưởng đến nhau.

- Cho phép kiểm thử các đồ hình phức tạp mà không cần phải nối dây cho mạng vật lý.
- Cho phép debug và chạy các phép kiểm thử trên các mạng giả lập lớn, sử dụng CLI.
- Hỗ trợ thiết lập các đồ hình tùy biến bất kỳ, gồm tập cơ bản các thông số đồ hình.
- Có thể đem các ứng dụng trên Mininet đi triển khai trên mạng thật với code hoàn toàn không cần thay đổi.
- Cung cấp Python API dễ dàng sử dụng và có khả năng mở rộng.

Mininet cho ta một phương pháp dễ dàng để thu được chính xác các đặc tính mạng và thử nghiệm các đồ hình tùy ý. Một trong những đặc điểm quan trọng nhất của mininet là chương trình dùng cho bộ điều khiển (controller) dùng cho giả lập bằng mininet hoàn toàn có thể đem ra triển khai cho mạng thật mà không cần sửa đổi.

III. YÊU CẦU VỀ THIẾT BỊ

Để thực hiện bài thí nghiệm này cần một máy tính chạy Ubuntu 12.04 đã cài sẵn phần mềm mininet.

IV. TRÌNH TỰ THÍ NGHIỆM

Trong bài thí nghiệm, sinh viên sẽ tiến hành tạo topology bằng mininet, sau đó add flow-entry cho OpenvSwitch.

❑ 1. Tạo một topology đơn giản bằng Mininet

Chạy lệnh sau

```
sudo mn --topo single, 3 --controller remote
```

Mininet sẽ tạo ra topo gồm có một Open vSwitch (s1) nối với 3 host: h1, h2, h3.

Trên màn hình Terminal sẽ hiện ra thông tin của topo như sau:

```
user@machinename:~$ sudo mn --topo single,3 --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
```

```
*** Adding switches:
```

```
s1
```

```
*** Adding links:
```

```
(h1, s1) (h2, s1) (h3, s1)
```

```
*** Configuring hosts
```

```
h1 h2 h3
```

```
*** Starting controller
```

```
c0
```

```
*** Starting 1 switches
```

```
s1 ...
```

```
*** Starting CLI:
```

```
mininet>
```

Với lựa chọn **--controller remote** Mininet sẽ tạo ra các Open vSwitch với cài đặt địa chỉ controller mặc định tại: IP=127.0.0.1 (localhost), port=6633.

Các controller có nhiệm vụ điều khiển hoạt động của mạng do Mininet tạo ra, đảm bảo mạng hoạt động ổn định, chính xác. Tuy nhiên, do chúng ta chưa có một Controller nào nằm ở địa chỉ IP= 127.0.0.1, port=6633, nên mạng sẽ không thể hoạt động.

Test thử hoạt động của mạng bằng cách gõ lệnh pingall trên màn hình CLI của mininet, ta sẽ thấy các host không thể ping được với nhau, 100% gói sẽ bị mất.

```
mininet> pingall
```

```
*** Ping: testing ping reachability
```

```
h1 -> X X
```

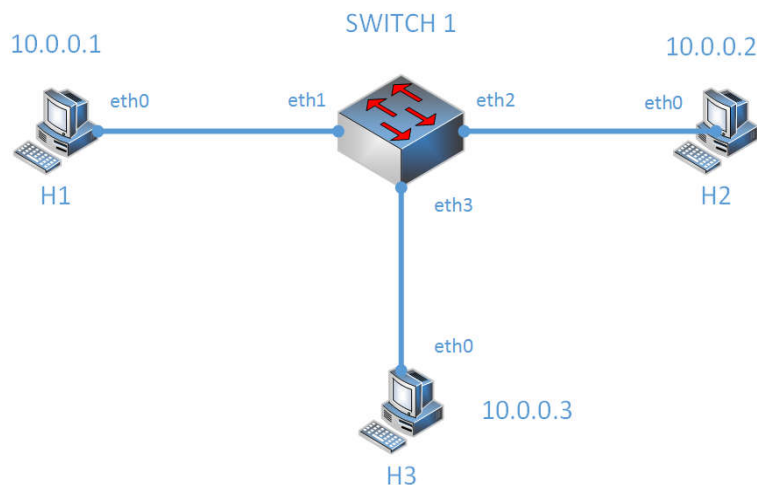
```
h2 -> X X
```

```
h3 -> X X
```

```
*** Results: 100% dropped (0/6 received)
```

```
mininet>
```

❑ 2. Thêm các flow-entry vào bảng flowtable của Open vSwitch



Hình 1. Topology 1 switch kết nối 3 host

Sinh viên tiến hành thêm các flow-entry cho switch **s1** để **host 1** có thể ping đến **host 3**.

Để hiểu rõ cơ chế ARP trong mạng, sinh viên xem lại bài thí nghiệm ARP.

Sinh viên tiến hành add các flow-entry cho switch **s1** theo thứ tự các bản tin trao đổi trong quá trình ping như sau (sinh viên mở một cửa sổ terminal mới để chạy các lệnh bên dưới).

- Đầu tiên, **host 1** sẽ gửi một bản tin ARP request cho **host 3** để lấy địa chỉ **MAC** của **host 3**. Vì vậy, ta cần thêm một flow-entry vào **switch 1** để nó forward gói tin này ra đúng port nối với **host 3**.

```
sudo ovs-ofctl add-flow s1 dl_type=0x0806,nw_dst=10.0.0.3, action=output:2
```

Lệnh **ovs-vsctl** sẽ thêm trực tiếp các flow-entry vào bảng flowtable của **switch 1** (0x0806: ARP)

Tương tự, với gói tin ARP reply từ **host 3** gửi về **host 1**, ta sẽ add flow-entry sau:

```
sudo ovs-ofctl add-flow s2 dl_type=0x0806,nw_dst=10.0.0.1,action=output:1
```

- Sau khi **host 1** nhận được gói tin ARP reply, nó sẽ tạo các gói tin ICMP request và ICMP reply. Tương tự như đối với các gói tin ARP ta sẽ add bằng các lệnh sau

```
sudo ovs-ofctl add-flow s1 dl_type=0x0800,nw_dst=10.0.0.3,action=output:2
sudo ovs-ofctl add-flow s1 dl_type=0x0800,nw_dst=10.0.0.1,action=output:1
```

(0x0800: ICMP)

Sau khi add xong tất cả các flow-entry, ta tiến hành ping từ host 1 đến host 3 bằng lệnh ***h1 ping h3*** trong giao diện CLI của mininet.

Để xác định đúng số port của switch ta chạy lệnh sau

```
sudo ovs-ofctl show s1
```

Toàn bộ thông tin của **switch 1**: dpid, kích thước table, kích thước buffer, trạng thái các port, tên các port sẽ được hiển thị trên terminal.

```
OFPT_FEATURES_REPLY (xid=0x1): ver:0x1, dpid:0000000000000001
n_tables:255, n_buffers:256
features: capabilities:0xc7, actions:0xffff
1(s1-eth1): addr:0e:cd:fc:f0:cb:1f
    config:    0
    state:     0
    current:   10GB-FD COPPER
2(s1-eth3): addr:f6:95:c5:60:e5:79
    config:    0
    state:     0
    current:   10GB-FD COPPER
3(s1-eth2): addr:ca:7f:49:ba:fb:d0
    config:    0
    state:     0
    current:   10GB-FD COPPER
LOCAL(s1): addr:6e:10:83:64:b5:47
    config:    PORT_DOWN
    state:     LINK_DOWN
OFPT_GET_CONFIG_REPLY (xid=0x3): frags=normal miss_send_len=0
```

V. KẾT LUẬN

Qua bài thí nghiệm này, sinh viên đã làm quen với phần mềm Mininet, biết cách tạo ra một topology đơn giản bằng mininet, hiểu cơ chế hoạt động của Open vSwitch.

VI. CÂU HỎI KIỂM TRA

1. Flow-entry và flowtable là gì?

.....

.....

.....

.....

2. Nêu các lệnh cần thiết để tạo flow-entry giúp tất cả các host có thể ping được với nhau?

.....

.....

.....

.....

3. Thời gian time-out của flow-entry trong flowtable là gì? Giá trị mặc định bằng bao nhiêu?

.....

.....

.....

.....