Nguyễn Thanh Quân - ntquan@fit.hcmus.edu.vn

Hướng dẫn sử dụng OpenFlow

Yêu cầu:

1. Download máy ảo Mininet (https://github.com/mininet/mininet/wiki/Mininet-VM-Images)

Nội dung:

1. SSH vào máy Mininet (chú ý -X)

```
ntquan@abc:~$ ssh -l mininet 192.168.167.132 -X
```

2. Chạy Mininet: Topology gồm 1 OpenFlow switch và 2 host

```
mininet@mininet-vm:~$ sudo mn --topo single,2 --mac --controller remote --switch ovsk
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

3. Chạy POX controller với **l2_pairs**

```
mininet@mininet-vm:~$ sudo ~/pox/pox.py forwarding.l2_pairs info.packet_dump samples.pretty_log log.level
--DEBUG
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
INFO:forwarding.l2_pairs:Pair-Learning switch running.
INFO:info.packet_dump:Packet dumper running
[core                ] POX 0.2.0 (carp) going up...
[core                ] Running on CPython (2.7.6/Mar 22 2014 22:59:56)
[core                ] Platform is Linux-3.13.0-24-generic-x86_64-with-Ubuntu-14.04-trusty
[core                ] POX 0.2.0 (carp) is up.
[openflow.of_01      ] Listening on 0.0.0.0:6633
[openflow.of_01      ] [00-00-00-00-00-01 1] connected ← switch kết nối vào controller
```

4. Trên Mininet, chạy h1 ping h2 với 1 gói, sau đó "dump-flows" để xem luật cài đặt

```
mininet@mininet-vm:~$ sudo mn --topo single,2 --mac --controller remote --switch ovsk
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
```

```
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ping h2 -c1
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=81.0 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 81.004/81.004/81.004/0.000 ms
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=2.218s, table=0, n_packets=1, n_bytes=98, idle_age=2, dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:2
cookie=0x0, duration=2.181s, table=0, n_packets=2, n_bytes=140, idle_age=2, dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:1
mininet>
```

Log của controller

```
mininet@mininet-vm:~$ sudo ~/pox/pox.py forwarding.l2_pairs info.packet_dump samples.pretty_log log.level
--DEBUG
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
INFO:forwarding.l2_pairs:Pair-Learning switch running.
INFO:info.packet_dump:Packet dumper running
[core                ] POX 0.2.0 (carp) going up...
[core                ] Running on CPython (2.7.6/Mar 22 2014 22:59:56)
[core                ] Platform is Linux-3.13.0-24-generic-x86_64-with-Ubuntu-14.04-trusty
[core                ] POX 0.2.0 (carp) is up.
[openflow.of_01      ] Listening on 0.0.0.0:6633
[openflow.of_01      ] [00-00-00-00-00-01 1] connected
[dump:00-00-00-00-00-01 ] [ethernet][arp]
[forwarding.l2_pairs    ] Installing 00:00:00:00:00:02 <-> 00:00:00:00:00:01
[dump:00-00-00-00-00-01 ] [ethernet][arp]
```

5. Chờ khoảng trên 5s, sau đó "dump-flows" lại, bạn sẽ thấy số lượng của "n_packets" tăng lên, tại sao?

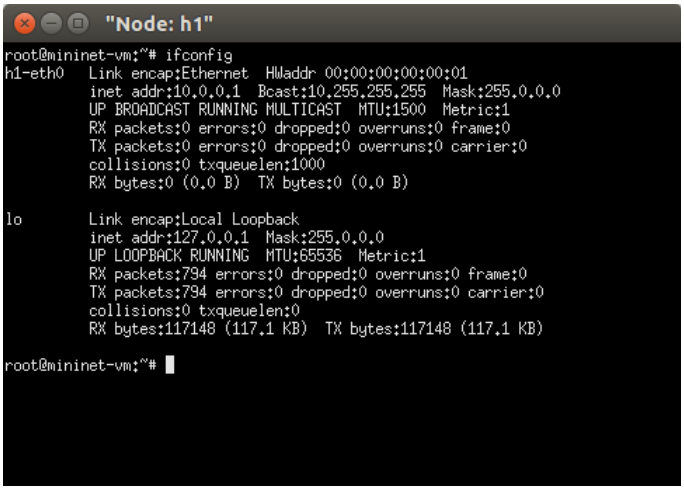
```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=2.218s, table=0, n_packets=1, n_bytes=98, idle_age=2, dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:2
 cookie=0x0, duration=2.181s, table=0, n_packets=2, n_bytes=140, idle_age=2, dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:1
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=5.982, table=0, n_packets=2, n_bytes=140, idle_age=107, dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:2
 cookie=0x0, duration=5.944s, table=0, n_packets=3, n_bytes=182, idle_age=107, dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:1
mininet>
```

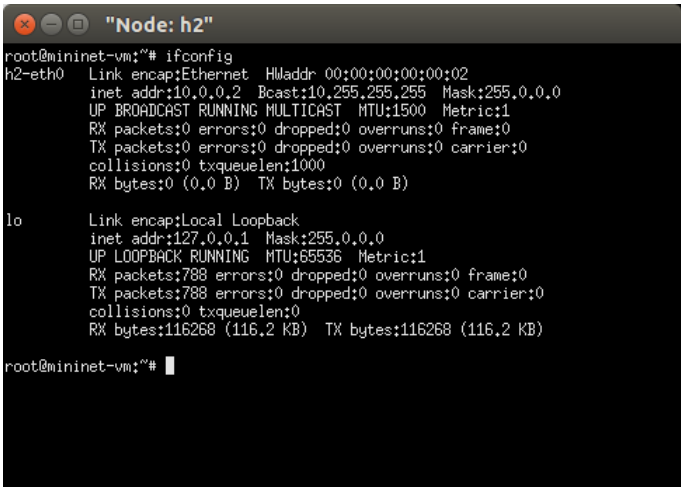
Giải thích: bằng cách debug số lượng gói tin trên h1 và h2

a. Chạy lại Mininet và remote vào máy h1, h2 bằng lệnh xterm (phải có -X khi ssh)

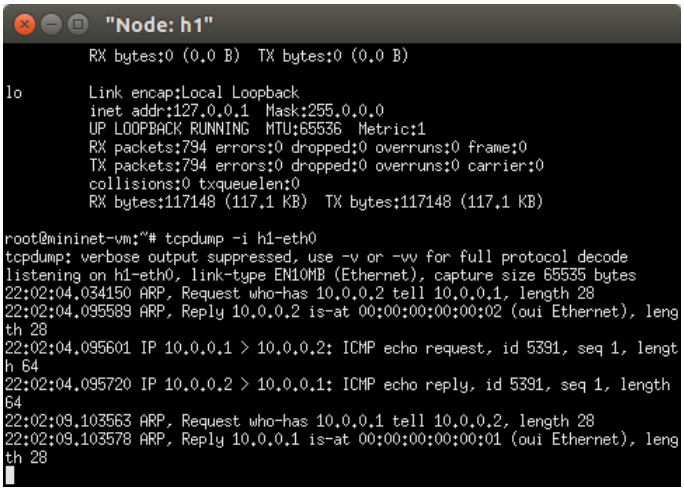
```
mininet@mininet-vm:~$ sudo mn --topo single,2 --mac --controller remote --switch ovsk
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet>
```

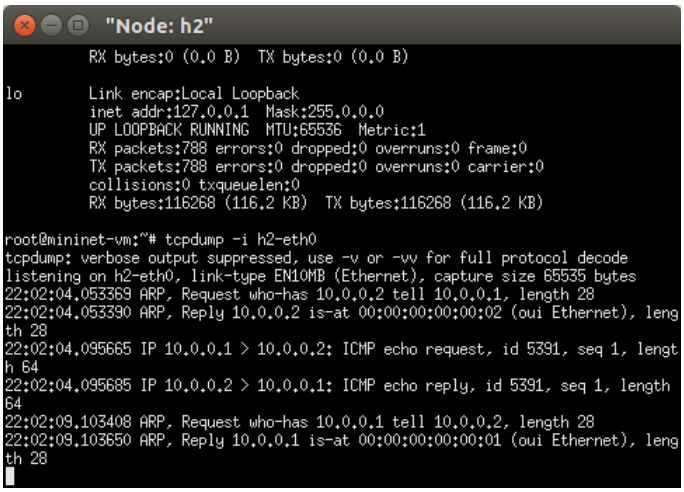
b. Có 2 màn hình control, mô phỏng 2 host h1 và h2





c. Chạy tcpdump trên 2 card mạng của “h1-eth0” và “h2-eth0”, sau đó trên Mininet, chạy lệnh “mininet>h1 ping h2 -c1” để coi số lượng gói tin mà 2 host này gửi





d. Debug trên OpenFlow switch “s1”

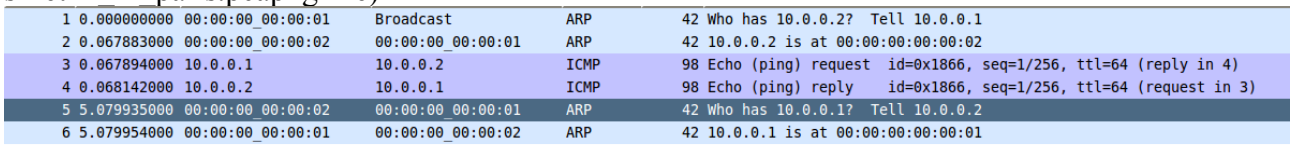
```
root@mininet-vm:~# ovs-ofctl snoop s1
OFPT_ECHO_REQUEST (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (xid=0x0): 0 bytes of payload

OFPT_PACKET_IN (xid=0x0): total_len=42 in_port=1 (via no_match) data_len=42 buffer=0x00000100
arp,metadata=0,in_port=0,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=ff:ff:ff:ff:ff:ff,arp_spa=10.0.
0.1,arp_tpa=10.0.0.2,arp_op=1,arp_sha=00:00:00:00:00:01,arp_tha=00:00:00:00:00:00
OFPT_PACKET_OUT (xid=0x7): in_port=1 actions=FLOOD buffer=0x00000100
OFPT_PACKET_IN (xid=0x0): total_len=42 in_port=2 (via no_match) data_len=42 buffer=0x00000101
arp,metadata=0,in_port=0,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,arp_spa=
10.0.0.2,arp_tpa=10.0.0.1,arp_op=2,arp_sha=00:00:00:00:00:02,arp_tha=00:00:00:00:00:01
2017-03-02T06:09:49Z|00001|ofp_util|INFO|normalization changed ofp_match, details:
2017-03-02T06:09:49Z|00002|ofp_util|INFO| pre:
dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=0.0.0.0,nw_dst=0.0.0.0,nw_proto=0,nw_tos=
0,tp_src=0,tp_dst=0
2017-03-02T06:09:49Z|00003|ofp_util|INFO|post: dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02
OFPT_FLOW_MOD (xid=0x8): ADD dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02
actions=output:2
2017-03-02T06:09:49Z|00004|ofp_util|INFO|normalization changed ofp_match, details:
2017-03-02T06:09:49Z|00005|ofp_util|INFO| pre:
dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=0.0.0.0,nw_dst=0.0.0.0,nw_proto=0,nw_tos=
0,tp_src=0,tp_dst=0
2017-03-02T06:09:49Z|00006|ofp_util|INFO|post: dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01
OFPT_FLOW_MOD (xid=0x9): ADD dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 buf:0x101
```

```
actions=output:1

OFPT_ECHO_REQUEST (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (xid=0x0): 0 bytes of payload
OFPT_ECHO_REQUEST (xid=0x0): 0 bytes of payload
OFPT_ECHO_REPLY (xid=0x0): 0 bytes of payload
```

e. Bắt gói tin trên card "s1-eth1" để xem tại sao ARP request thứ 2 xử lý như thế nào (xem s1-eth1_l2_pairs.pcapng file)



1	0.000000000	00:00:00_00:00:01	Broadcast	ARP	42 Who has 10.0.0.2? Tell 10.0.0.1
2	0.067883000	00:00:00_00:00:02	00:00:00_00:00:01	ARP	42 10.0.0.2 is at 00:00:00:00:00:02
3	0.067894000	10.0.0.1	10.0.0.2	ICMP	98 Echo (ping) request id=0x1866, seq=1/256, ttl=64 (reply in 4)
4	0.068142000	10.0.0.2	10.0.0.1	ICMP	98 Echo (ping) reply id=0x1866, seq=1/256, ttl=64 (request in 3)
5	5.079935000	00:00:00_00:00:02	00:00:00_00:00:01	ARP	42 Who has 10.0.0.1? Tell 10.0.0.2
6	5.079954000	00:00:00_00:00:01	00:00:00_00:00:02	ARP	42 10.0.0.1 is at 00:00:00:00:00:01

Lập trình Controller with POX:

Tham khảo: http://sdnhub.org/tutorials/pox

Code của l2_pairs.py xem tại đường dẫn "/home/mininet/pox/pox/forwarding". Lưu ý khi l2_pairs cài đặt 2 FLOW_MOD, với FLOW_MOD thứ 2, nó cũng yêu cầu OpenFlow switch xử lý gói tin này theo luật thứ 2. (GV không thấy nói chỗ này :P)

```
def _handle_PacketIn (event):
  packet = event.parsed

  # Learn the source
  table[(event.connection,packet.src)] = event.port

  dst_port = table.get((event.connection,packet.dst))

  if dst_port is None:
    # We don't know where the destination is yet.  So, we'll just
    # send the packet out all ports (except the one it came in on!)
    # and hope the destination is out there somewhere. :)
    msg = of.ofp_packet_out(data = event.ofp)
    msg.actions.append(of.ofp_action_output(port = all_ports))
    event.connection.send(msg)
    log.debug("FLOOD ---->>>>>>>>>>>>>>>>")
  else:
    # Since we know the switch ports for both the source and dest
    # MACs, we can install rules for both directions.
    msg = of.ofp_flow_mod()
    msg.match.dl_dst = packet.src
    msg.match.dl_src = packet.dst
    msg.actions.append(of.ofp_action_output(port = event.port))
    event.connection.send(msg)

    # This is the packet that just came in -- we want to
    # install the rule and also resend the packet.
    msg = of.ofp_flow_mod()
    msg.data = event.ofp # Forward the incoming packet ← lúc đọc code GV không để ý
    msg.match.dl_src = packet.src
    msg.match.dl_dst = packet.dst
    msg.actions.append(of.ofp_action_output(port = dst_port))
    event.connection.send(msg)

    log.debug("Installing %s <-> %s" % (packet.src, packet.dst))
```

Trong thư mục "/home/mininet/pox/pox/forwarding" có nhiều ứng dụng đã lập trình sẵn. Sinh viên vui lòng xem qua như:

- hub.py: cài đặt OpenFlow switch hoạt động như một thiết bị HUB

```
mininet@mininet-vm:~$ sudo ~/pox/pox.py forwarding.hub info.packet_dump samples.pretty_log log.level
--DEBUG
```

- l2_learning.py: cài đặt OpenFlow switch hoạt động như một thiết bị SWITCH, sinh viên đọc qua và so sánh sự khác biệt với "l2_pairs.py" như thế nào. Trong file "l2_learning.py" có thông tin là "idle_timeout", "hard_timeout" là gì? vui lòng đọc trong "Tổng quan về OpenFlow" file.

```
mininet@mininet-vm:~$ sudo ~/pox/pox.py forwarding.l2_learning info.packet_dump samples.pretty_log log.level
--DEBUG
```

Lập trình Controller with RYU: (Topology Viewer)

Tham khảo: http://sdnhub.org/tutorials/ryu/

Sinh viên cài đặt hệ điều hành Ubuntu (ví dụ), sau đó cài RYU controller theo link: "https://osrg.github.io/ryu/"

```
# git clone git://github.com/osrg/ryu.git
# cd ryu; python ./setup.py install
```

1. Giả sử máy cài RYU controller có IP là "192.168.167.1", tạo Mininet

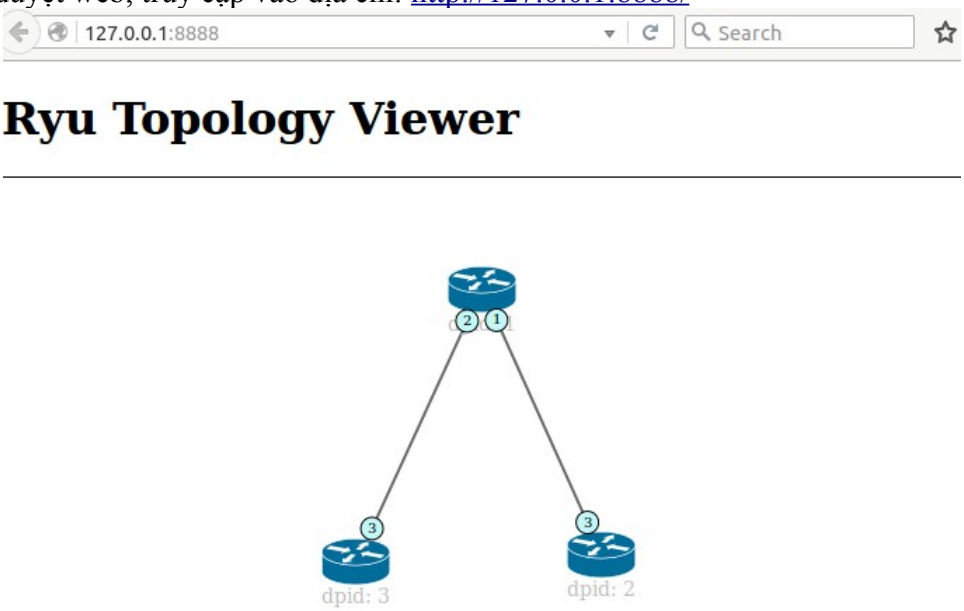
```
mininet@mininet-vm:~$ sudo mn --topo tree,2,2 --mac --controller remote,ip=192.168.167.1 --switch ovsk
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.167.1:6633
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, s2) (s1, s3) (s2, h1) (s2, h2) (s3, h3) (s3, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet>
```

2. Trên máy RYU controller, chạy như sau

```
root@tma-spn:~# ryu-manager --observe-links ryu/ryu/app/gui_topology/gui_topology.py --wsapi-port 8888
loading app ryu/ryu/app/gui_topology/gui_topology.py
loading app ryu.app.rest_topology
loading app ryu.app.ws_topology
loading app ryu.app.ofctl_rest
loading app ryu.controller.ofp_handler
instantiating app None of Switches
creating context switches
instantiating app None of DPSet
```

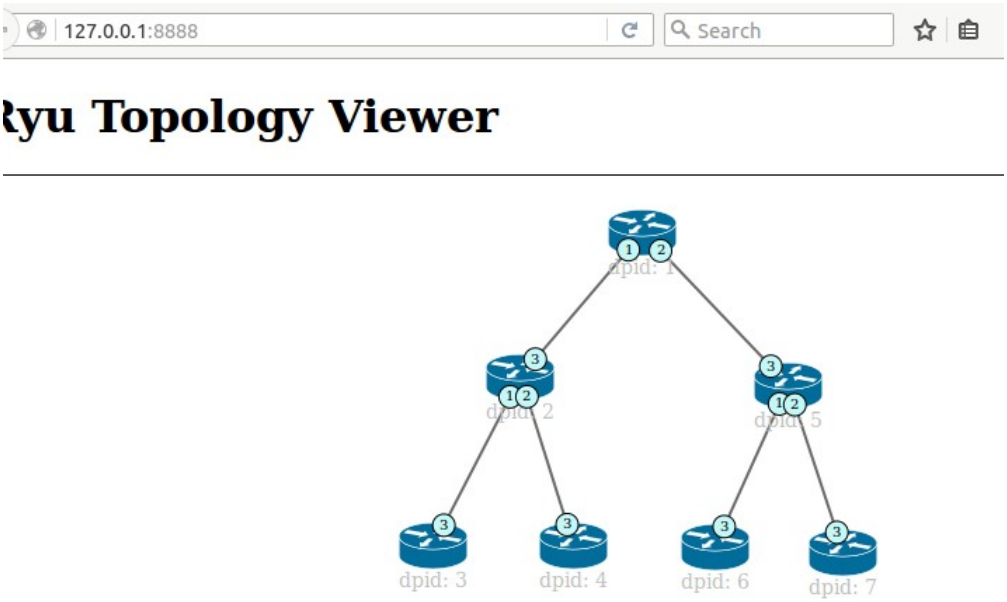
```
creating context dpset
creating context wsgi
instantiating app ryu.app.ws_topology of WebSocketTopology
instantiating app ryu.app.rest_topology of TopologyAPI
instantiating app ryu/ryu/app/gui_topology/gui_topology.py of GUIServerApp
instantiating app ryu.app.ofctl_rest of RestStatsApi
instantiating app ryu.controller.ofp_handler of OFPHandler
(25274) wsgi starting up on http://0.0.0.0:8888/
(25274) accepted ('127.0.0.1', 59689)
```

3. Mở trình duyệt web, truy cập vào địa chỉ: http://127.0.0.1:8888/



4. Tạo một Topology khác với Mininet và xem Topology

```
mininet@mininet-vm:~$ sudo mn --topo tree,depth=3 --mac --controller remote,ip=192.168.167.1 --switch ovsk
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7
*** Adding links:
(s1, s2) (s1, s5) (s2, s3) (s2, s4) (s3, h1) (s3, h2) (s4, h3) (s4, h4) (s5, s6) (s5, s7) (s6, h5) (s6, h6) (s7, h7) (s7, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Starting CLI:
mininet>
```


127.0.0.1:8888
Search
Ryu Topology Viewer
dpid: 3
dpid: 4
dpid: 6
dpid: 7