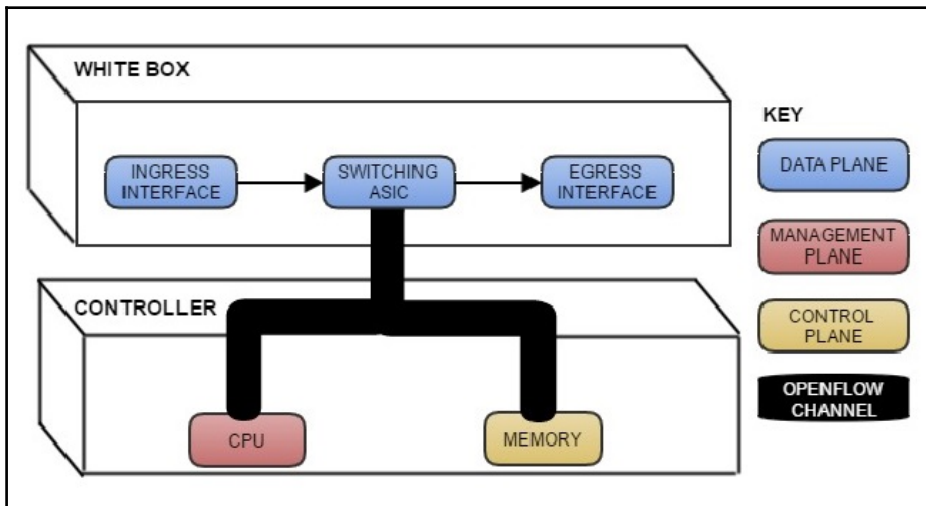
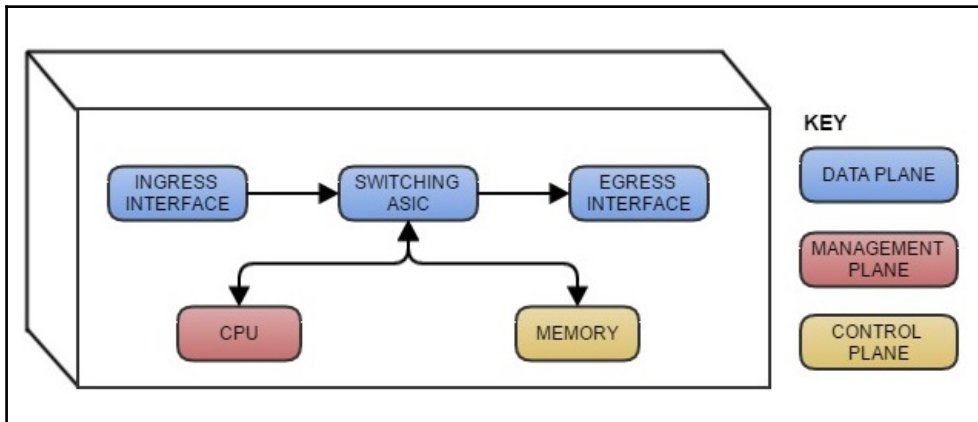
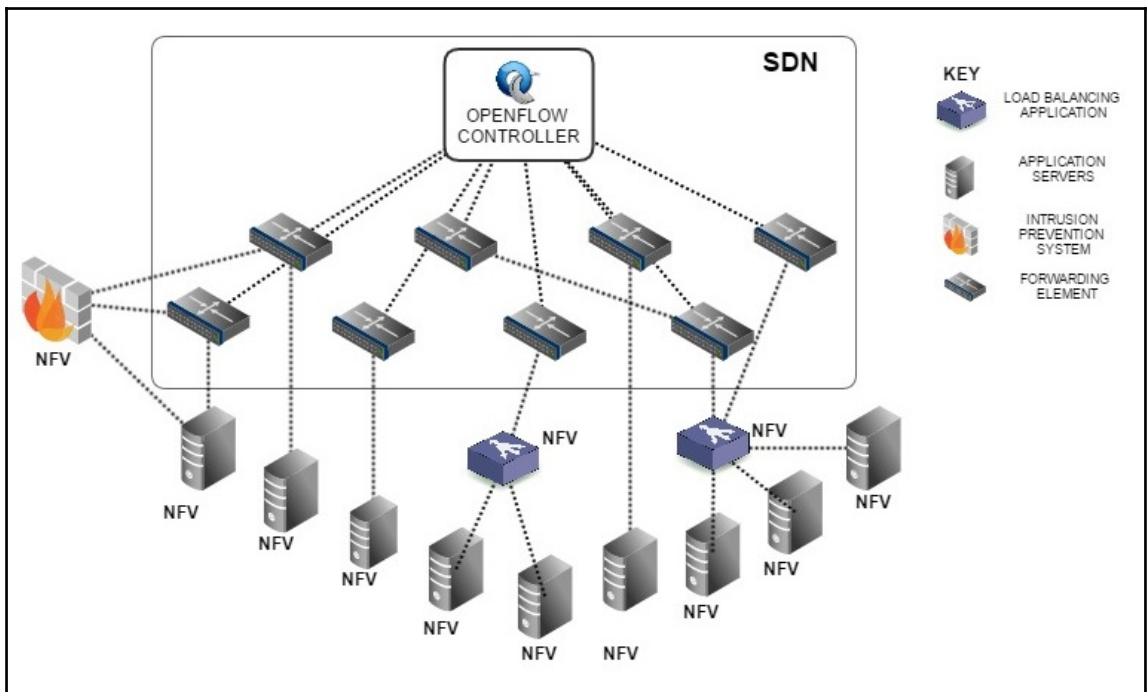
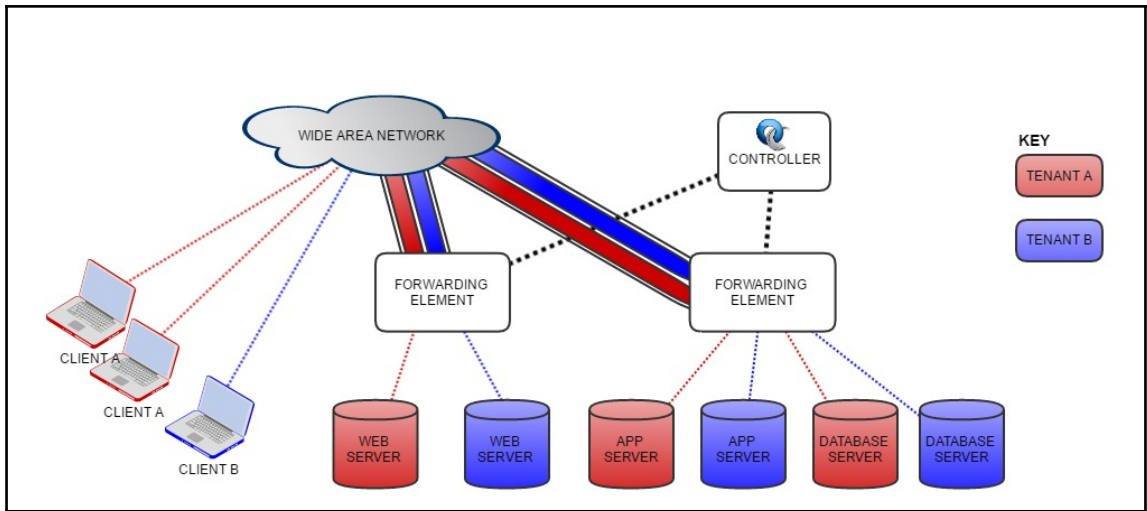


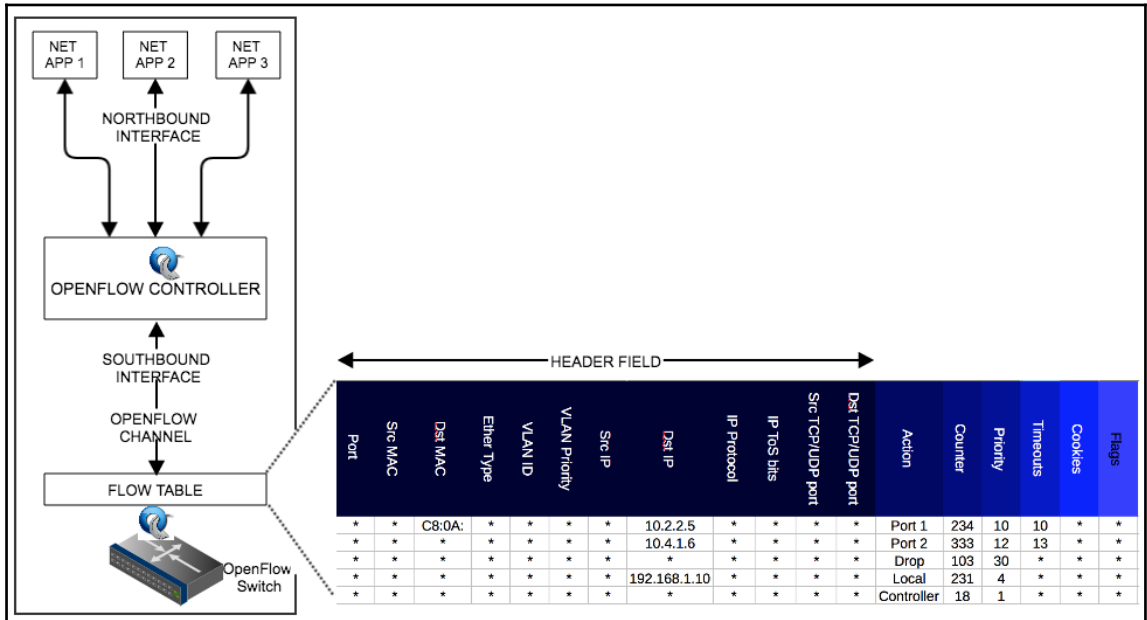
# Graphics Bundle

## Chapter 1: Software-Defined Networks

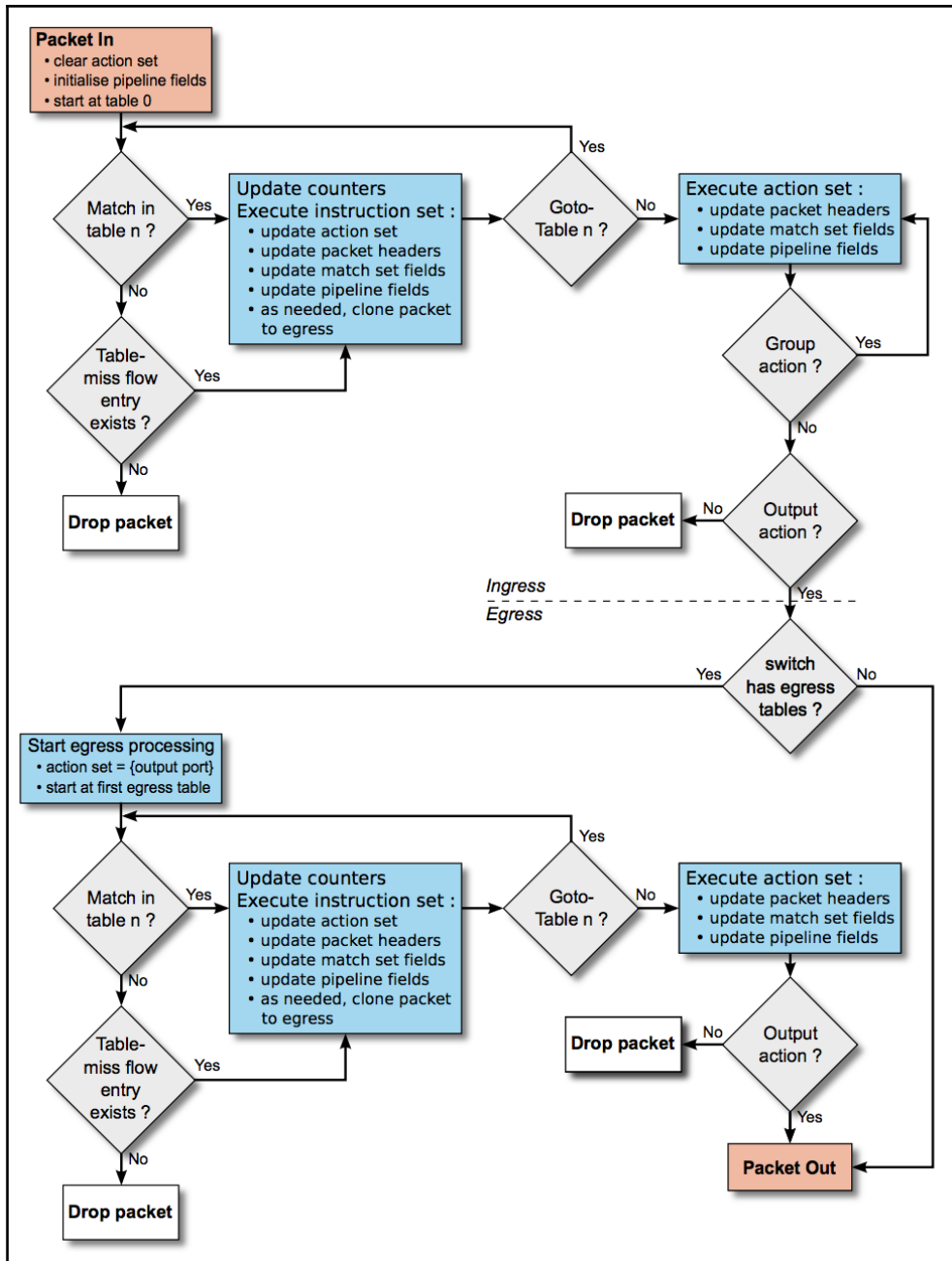




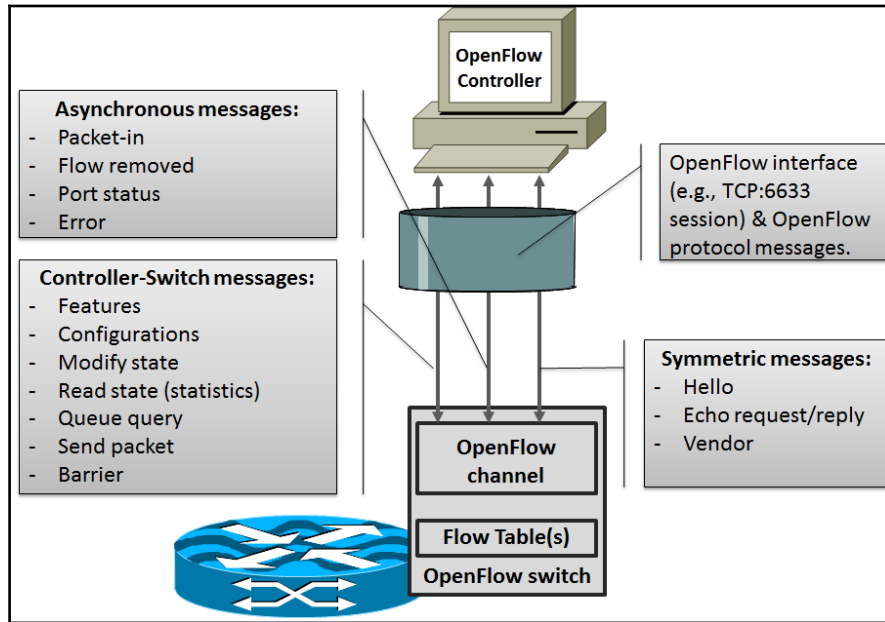
# Chapter 2: Introducing OpenFlow

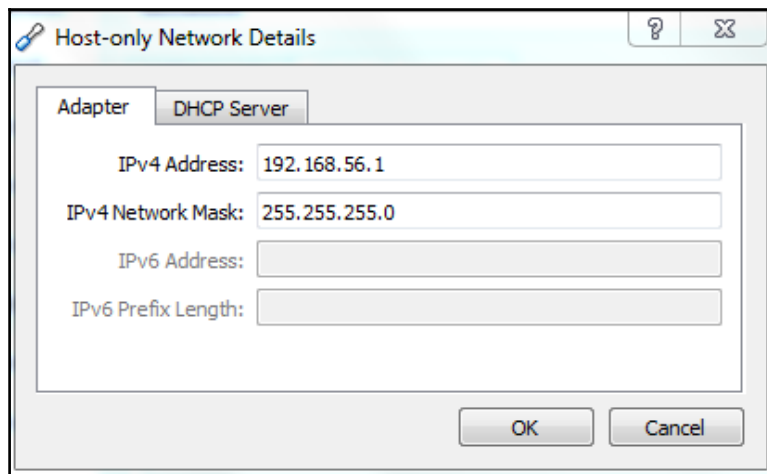
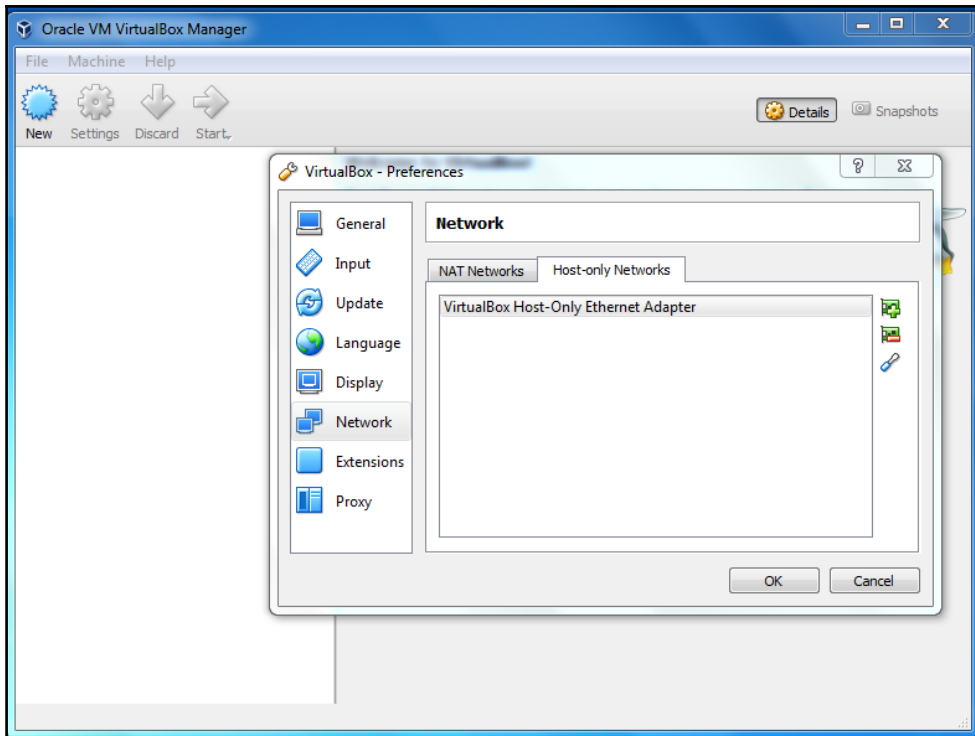


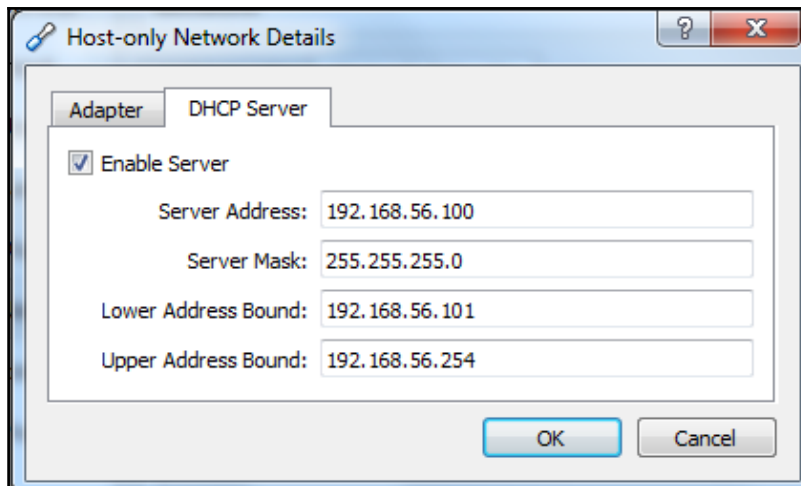
Per Port			Per Meter		
Received Packets	64	<i>Required</i>	Flow Count	32	<i>Optional</i>
Transmitted Packets	64	<i>Required</i>	Input Packet Count	64	<i>Optional</i>
Received Bytes	64	<i>Optional</i>	Input Byte Count	64	<i>Optional</i>
Transmitted Bytes	64	<i>Optional</i>	Duration (seconds)	32	<i>Required</i>
Receive Drops	64	<i>Optional</i>	Duration (nanoseconds)	32	<i>Optional</i>
Transmit Drops	64	<i>Optional</i>	Per Flow Entry		
Receive Errors	64	<i>Optional</i>	Received Packets	64	<i>Optional</i>
Transmit Errors	64	<i>Optional</i>	Received Bytes	64	<i>Optional</i>
Receive Frame Alignment Errors	64	<i>Optional</i>	Duration (seconds)	32	<i>Required</i>
Receive Overrun Errors	64	<i>Optional</i>	Duration (nanoseconds)	32	<i>Optional</i>
Receive CRC Errors	64	<i>Optional</i>	Per Flow Table		
Collisions	64	<i>Optional</i>	Reference Count (active entries)	32	<i>Required</i>
Duration (seconds)	32	<i>Required</i>	Packet Lookups	64	<i>Optional</i>
Duration (nanoseconds)	32	<i>Optional</i>	Packet Matches	64	<i>Optional</i>
Per Meter Band			Per Group		
In Band Packet Count	64	<i>Optional</i>	Reference Count (flow entries)	32	<i>Optional</i>
In Band Byte Count	64	<i>Optional</i>	Packet Count	64	<i>Optional</i>
Per Queue			Byte Count	64	<i>Optional</i>
Transmit Packets	64	<i>Required</i>	Duration (seconds)	32	<i>Required</i>
Transmit Bytes	64	<i>Optional</i>	Duration (nanoseconds)	32	<i>Optional</i>
Transmit Overrun Errors	64	<i>Optional</i>	Per Group Bucket		
Duration (seconds)	32	<i>Required</i>	Packet Count	64	<i>Optional</i>
Duration (nanoseconds)	32	<i>Optional</i>	Byte Count	64	<i>Optional</i>



# Chapter 3: Implementing the OpenFlow Switch





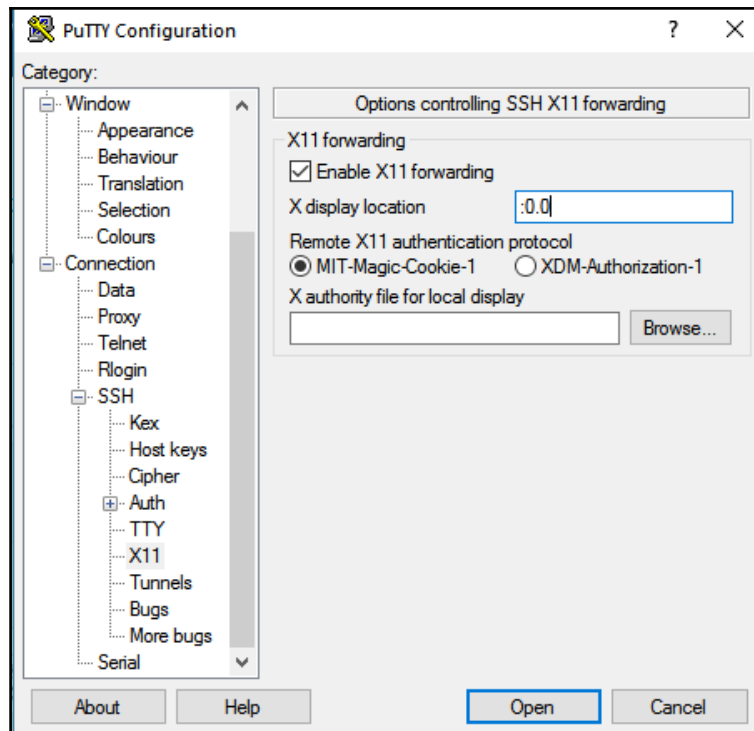


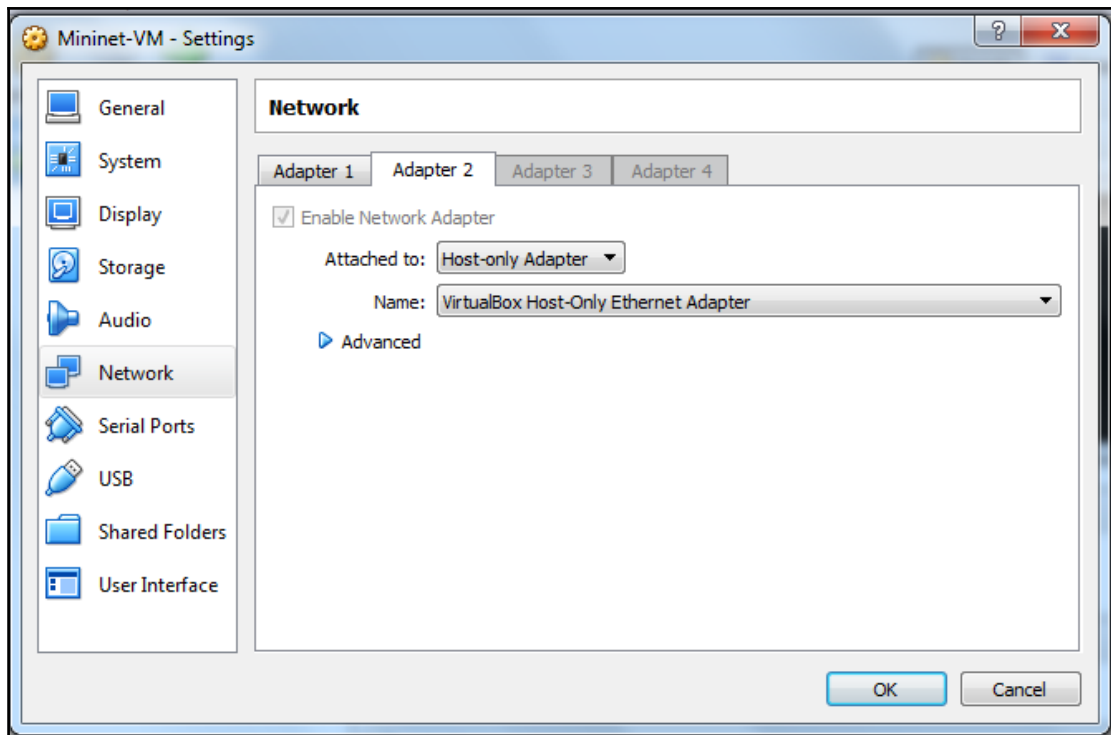
## Appliance to import

VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

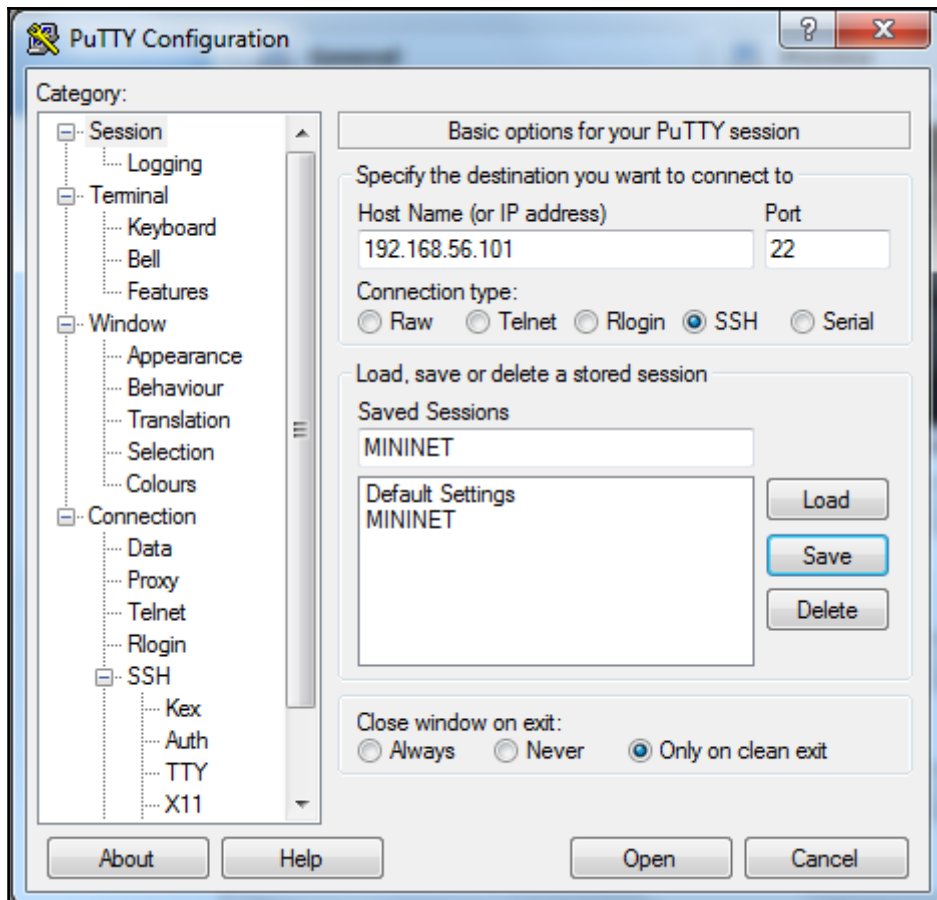








```
mininet@mininet-vm: ~  
mininet@mininet-vm:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:fd:2f:9c  
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:23222 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:28155 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:4479746 (4.4 MB)  TX bytes:41723488 (41.7 MB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:31970 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:31970 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:24676976 (24.6 MB)  TX bytes:24676976 (24.6 MB)  
  
mininet@mininet-vm:~$
```



The image shows a Wireshark capture of network traffic and a terminal window. The terminal window displays the configuration of a network topology and the execution of ping tests.

**Terminal Output:**

```

mininet@mininetvm ~
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 -c 1 h2
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=1.73 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.733/1.733/1.733/0.000 ms
mininet> h1 ping -c 1 h2
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=0.236 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.236/0.236/0.236/0.000 ms
mininet>

```

**Wireshark Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
467	4.339562000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_request
468	4.340450000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_reply
984	9.339597000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_request
985	9.340279000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_reply
1426	14.340940000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_request
1427	14.340940000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_reply
1483	19.339595000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_request
1484	19.340431000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_reply
1511	24.339593000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_request
1512	24.340705000	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_reply

**Wireshark Packet Details:**

- Frame 467: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
- Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
- Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
- Transmission Control Protocol, Src Port: 35902 (35902), Dst Port: openflow (6653), Seq: 1, Ack: 1, Len: 8
- OpenFlow (LOXI)

**Wireshark Packet Bytes:**

```

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 45 c0 .....E.
0010 00 3c c2 fa 40 00 40 06 78 ff 7f 00 00 01 7f 00 <.@.X.....
0020 00 01 8c 3e 10 fd 07 86 8f 6f 31 d2 e2 2a 80 18 >.>...e1.*.
0030 00 56 fe 30 00 00 01 01 00 0a 00 02 7d b2 00 02 V.0.....)
0040 78 00 01 02 00 08 00 00 00 00

```

**Capturing from lo (loopback) [Wireshark 1.8.2]**

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: of Restart the running live capture Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
53	3.565238000	127.0.0.1	127.0.0.1	OFF	7	Hello (SM) (8B)
55	3.566078000	127.0.0.1	127.0.0.1	OFF	7	Hello (SM) (8B)
57	3.566188000	127.0.0.1	127.0.0.1	OFF	7	Features Request (CSM) (8B)
59	3.566234000	127.0.0.1	127.0.0.1	OFF	7	Set Config (CSM) (12B)
61	3.566905000	127.0.0.1	127.0.0.1	OFF	24	Features Reply (CSM) (176B)
63	3.613907000	::	ff02::1:ff89:e474	OFF+ICMP	16	Packet In (AM) (BufID=256) (96B) => Neighbor Solicitation for fe80::e4ad:afff:fe89:e474
65	3.614331000	127.0.0.1	127.0.0.1	OFF	16	Packet Out (CSM) (BufID=256) (102B)
67	3.657816000	::	ff02::1:ffe9:fb76	OFF+ICMP	16	Packet In (AM) (BufID=257) (96B) => Neighbor Solicitation for fe80::88e0:48ff:fe9:fb76
68	3.658149000	127.0.0.1	127.0.0.1	OFF	16	Packet Out (CSM) (BufID=257) (102B)
80	4.202156000	127.0.0.1	127.0.0.1	OFF	19	Port Status (AM) (64B)
100	4.613942000	Fe80::e4ad:afff:fe89:ff02::2	127.0.0.1	OFF+ICMP	15	Packet In (AM) (BufID=258) (88B) => Router Solicitation from e6:ad:af:89:e4:74
102	4.614407000	127.0.0.1	127.0.0.1	OFF	16	Packet Out (CSM) (BufID=258) (94B)
104	4.657931000	Fe80::88e0:48ff:fe9:ff02::2	127.0.0.1	OFF+ICMP	15	Packet In (AM) (BufID=259) (88B) => Router Solicitation from 8a:e0:48:e9:fb:76
105	4.658260000	127.0.0.1	127.0.0.1	OFF	16	Packet Out (CSM) (BufID=259) (94B)
130	7.606051000	Fe80::e4ad:afff:fe89:ff02::16	127.0.0.1	OFF+ICMP	17	Packet In (AM) (BufID=260) (108B) => Multicast Listener Report Message v2
131	7.606549000	127.0.0.1	127.0.0.1	OFF	18	Packet Out (CSM) (BufID=260) (114B)
143	8.622032000	Fe80::e4ad:afff:fe89:ff02::2	127.0.0.1	OFF+ICMP	15	Packet In (AM) (BufID=261) (88B) => Router Solicitation from e6:ad:af:89:e4:74
144	8.622512000	127.0.0.1	127.0.0.1	OFF	16	Packet Out (CSM) (BufID=261) (94B)
146	8.662076000	Fe80::88e0:48ff:fe9:ff02::2	127.0.0.1	OFF+ICMP	15	Packet In (AM) (BufID=262) (88B) => Router Solicitation from 8a:e0:48:e9:fb:76
147	8.662541000	127.0.0.1	127.0.0.1	OFF	16	Packet Out (CSM) (BufID=262) (94B)
159	9.654141000	Fe80::88e0:48ff:fe9:ff02::16	127.0.0.1	OFF+ICMP	17	Packet In (AM) (BufID=263) (108B) => Multicast Listener Report Message v2
160	9.654618000	127.0.0.1	127.0.0.1	OFF	18	Packet Out (CSM) (BufID=263) (114B)
218	12.499251000	127.0.0.1	127.0.0.1	OFF	19	Port Status (AM) (64B)

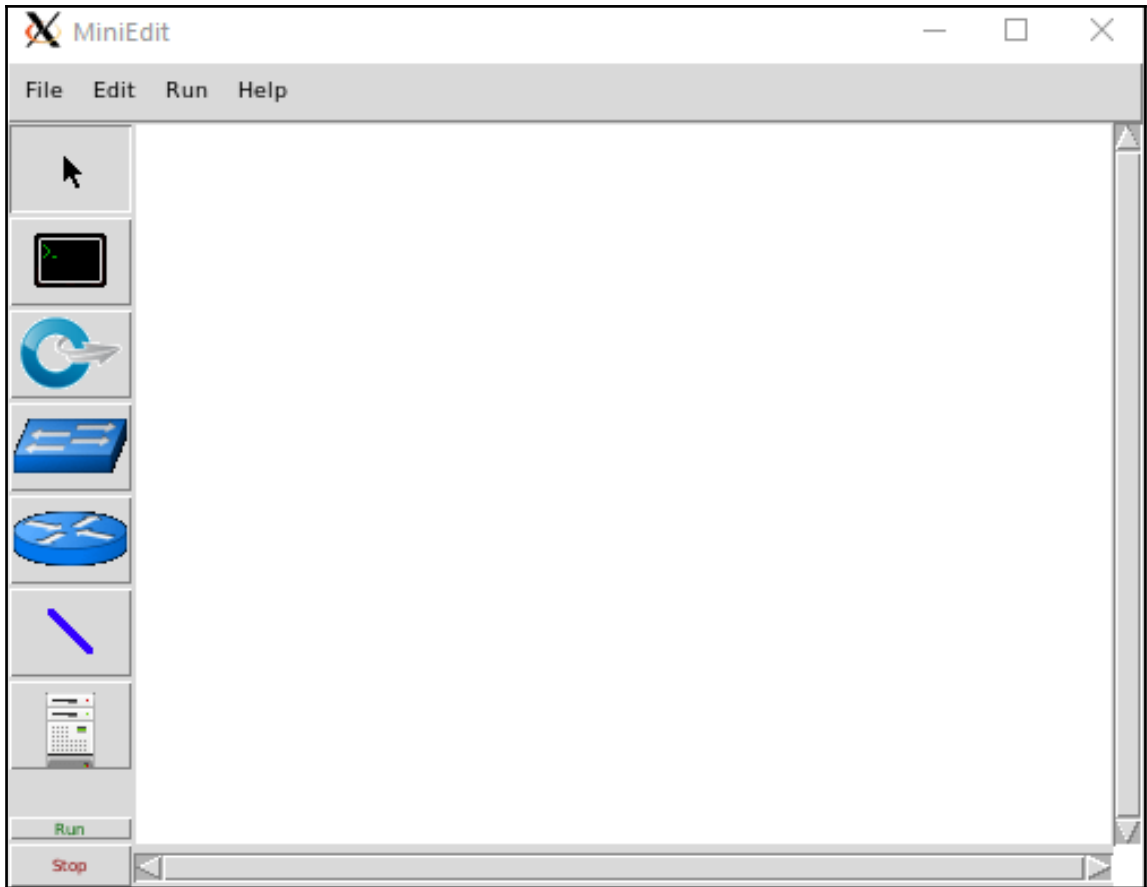
▶ Frame 53: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0  
 ▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)  
 ▶ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)  
 ▶ Transmission Control Protocol, Src Port: 33761 (33761), Dst Port: 6633 (6633), Seq: 1, Ack: 1, Len: 8  
 ▶ OpenFlow Protocol

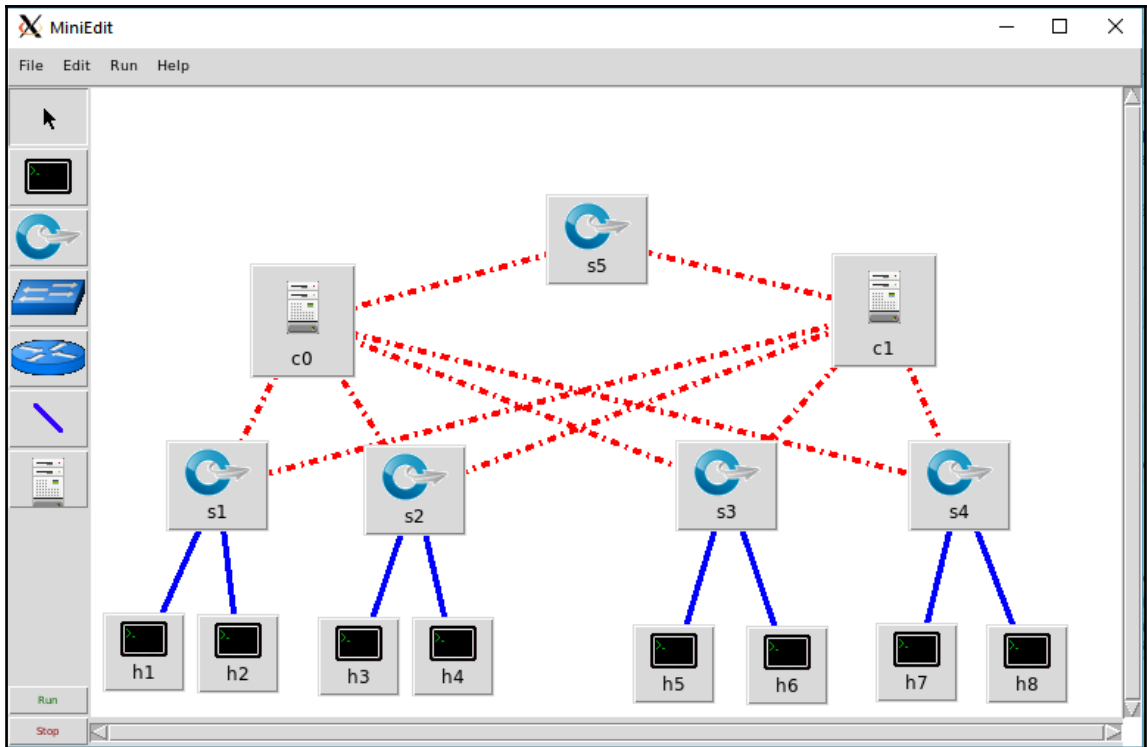
```

0000 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00 .....E.
0010 00 3c a1 5f 40 00 40 06 9b 5a 7f 00 00 01 7f 00 <._@.@.Z.....
0020 00 01 83 a1 19 a9 9a 38 c2 91 a2 23 56 a4 80 18 .....8...#V...
0030 00 41 fe 30 00 00 01 01 08 0a 00 0f 63 cd 00 0f .A.O.....c...
0040 63 cd 01 00 00 08 00 00 00 08
  
```

lo (loopback): <live capture in progress> File: /tmp... Packets: 1858 Displayed: 23 Marked: 0

No.	Time	Source	Destination	Protocol	Length	Info
159	12.344214000	127.0.0.1	127.0.0.1	OFF	74	Echo Reply (SM) (8B)
185	17.343212000	127.0.0.1	127.0.0.1	OFF	74	Echo Request (SM) (8B)
186	17.343649000	127.0.0.1	127.0.0.1	OFF	74	Echo Reply (SM) (8B)
206	20.550900000	52:b1:07:ad:72:8f	Broadcast	OFF+ARP	126	Packet In (AM) (BufID=265) (60B) => Who has 10.0.0.2? Tell 10.0.0.1
207	20.551428000	127.0.0.1	127.0.0.1	OFF	132	Packet Out (CSM) (BufID=265) (66B)
209	20.552121000	62:f7:71:45:5c:21	52:b1:07:ad:72:8f	OFF+ARP	126	Packet In (AM) (BufID=266) (60B) => 10.0.0.2 is at 62:f7:71:45:5c:21
210	20.552584000	127.0.0.1	127.0.0.1	OFF	146	Flow Mod (CSM) (80B)
211	20.553800000	10.0.0.2	10.0.0.2	OFF+ICMP	182	Packet In (AM) (BufID=267) (116B) => Echo (ping) request id=0x0b76, seq=1/256, ttl=64
212	20.553659000	127.0.0.1	127.0.0.1	OFF	146	Flow Mod (CSM) (80B)
238	25.342724000	127.0.0.1	127.0.0.1	OFF	74	[TCP ACKED unseen segment] Echo Request (SM) (8B)
239	25.343129000	127.0.0.1	127.0.0.1	OFF	74	[TCP ACKED unseen segment] [TCP Previous segment not captured] Echo Reply (SM) (8B)
249	25.557556000	62:f7:71:45:5c:21	52:b1:07:ad:72:8f	OFF+ARP	126	Packet In (AM) (BufID=269) (60B) => Who has 10.0.0.1? Tell 10.0.0.2
250	25.558169000	127.0.0.1	127.0.0.1	OFF	146	Flow Mod (CSM) (80B)
252	25.558613000	52:b1:07:ad:72:8f	62:f7:71:45:5c:21	OFF+ARP	126	Packet In (AM) (BufID=270) (60B) => 10.0.0.1 is at 52:b1:07:ad:72:8f
253	25.559469000	127.0.0.1	127.0.0.1	OFF	146	Flow Mod (CSM) (80B)
324	30.341971000	127.0.0.1	127.0.0.1	OFF	74	Echo Request (SM) (8B)
325	30.342391000	127.0.0.1	127.0.0.1	OFF	74	Echo Reply (SM) (8B)
349	35.344656000	127.0.0.1	127.0.0.1	OFF	74	Echo Request (SM) (8B)





**Controller Details**

Name:

Controller Port:

Controller Type:

Protocol:

Remote/In-Band Controller

IP Address:



MiniEdit

Hostname: s2 External Interface: Add

DPID:

External Interfaces

Enable NetFlow:

Enable sFlow:

Switch Type: Default

IP Address:

DPCTL port:

Start Command:

Stop Command:

OK Cancel

MiniEdit

Properties | VLAN Interfaces | External Interfaces | Private Directories

Hostname: h4

IP Address:

Default Route:

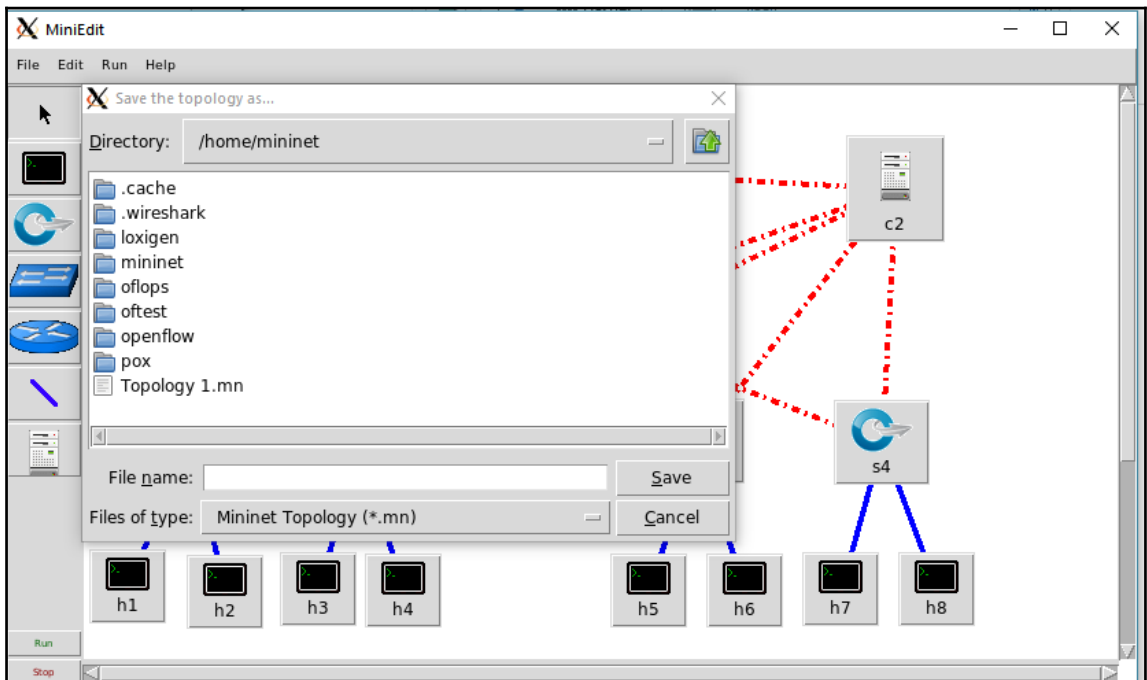
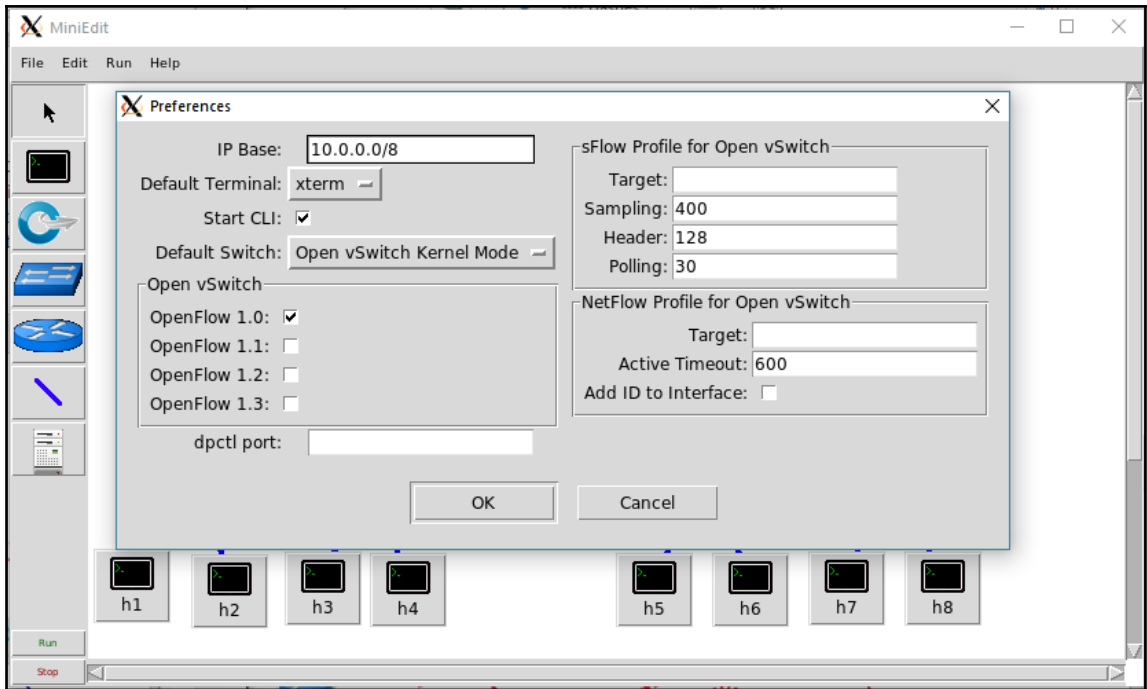
Amount CPU:  host

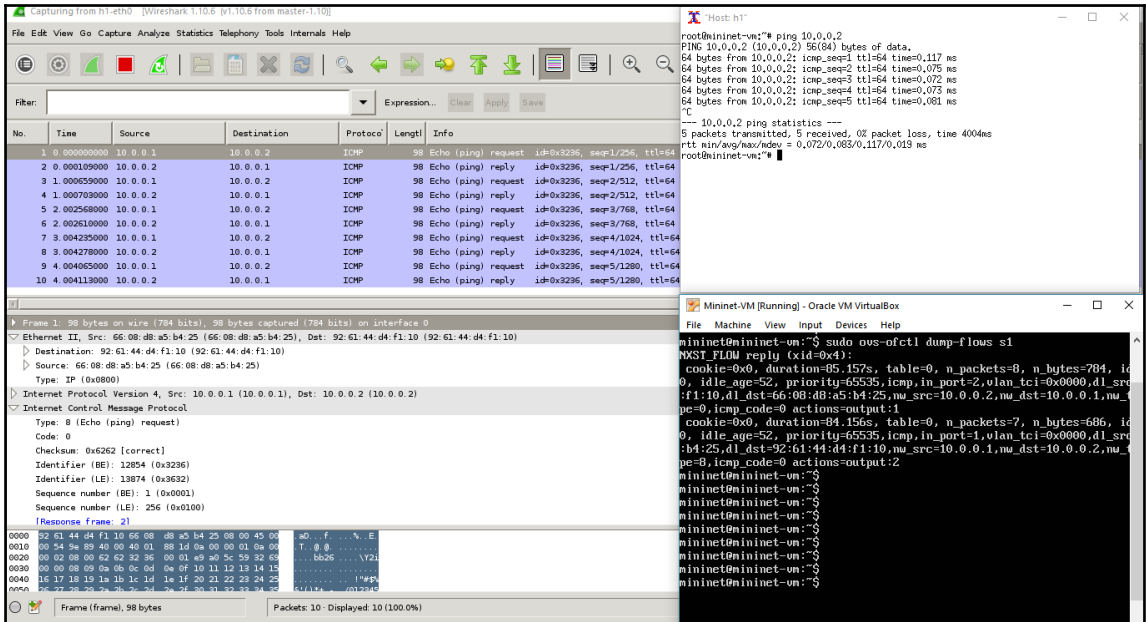
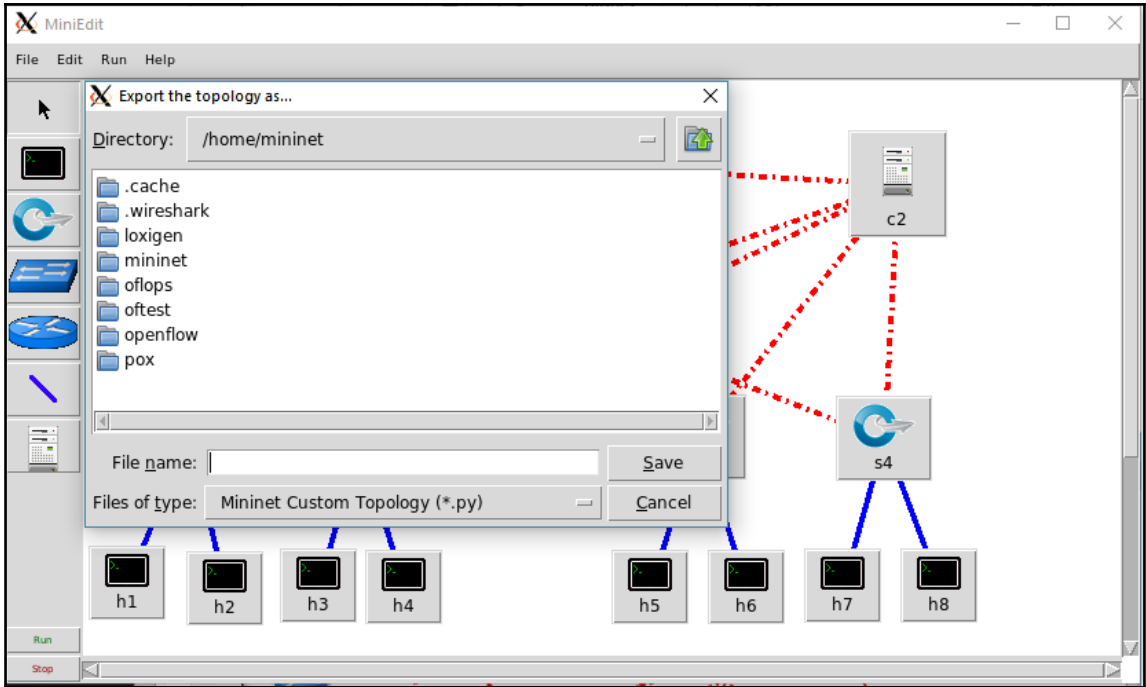
Cores:

Start Command:

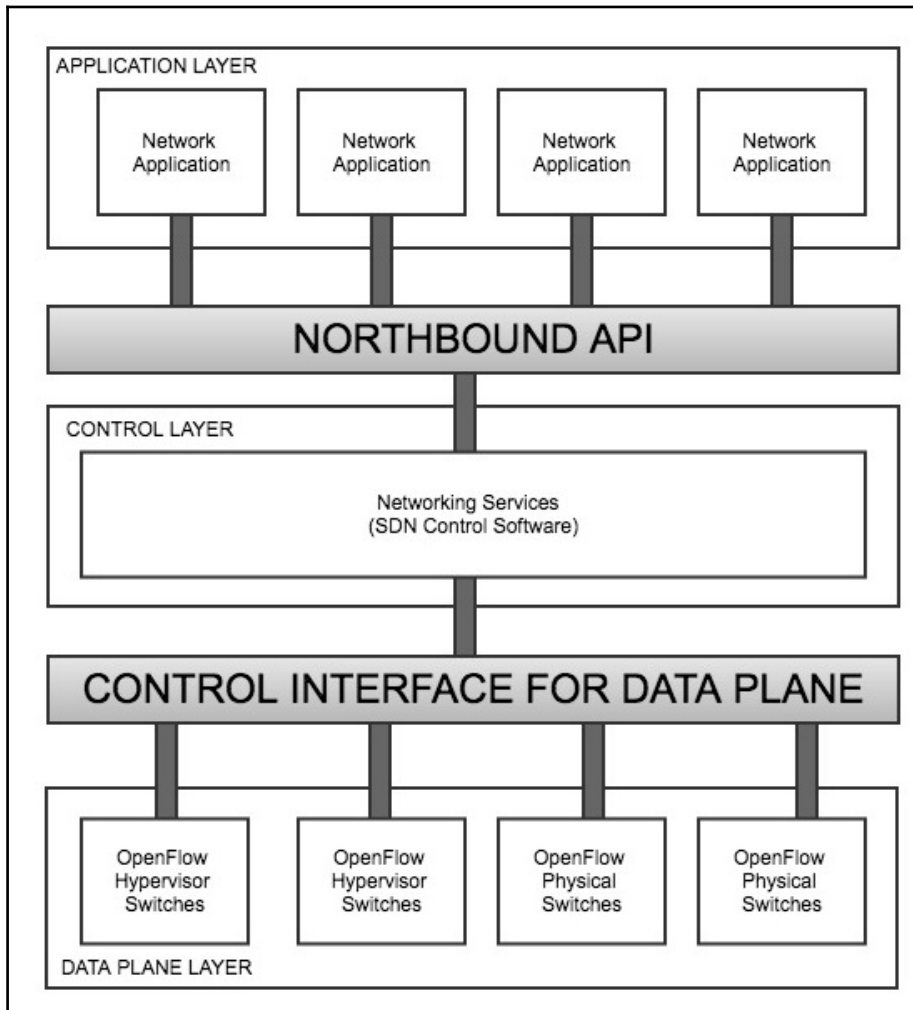
Stop Command:

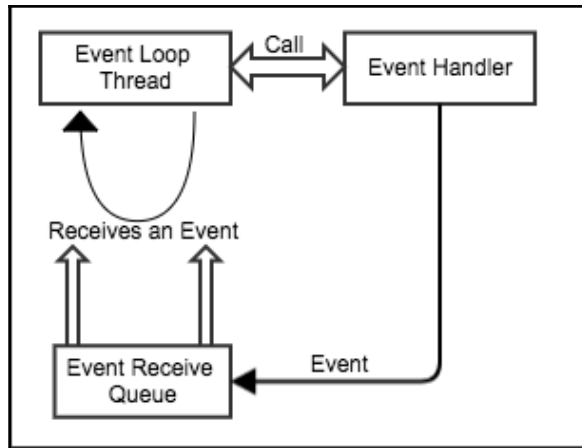
OK Cancel



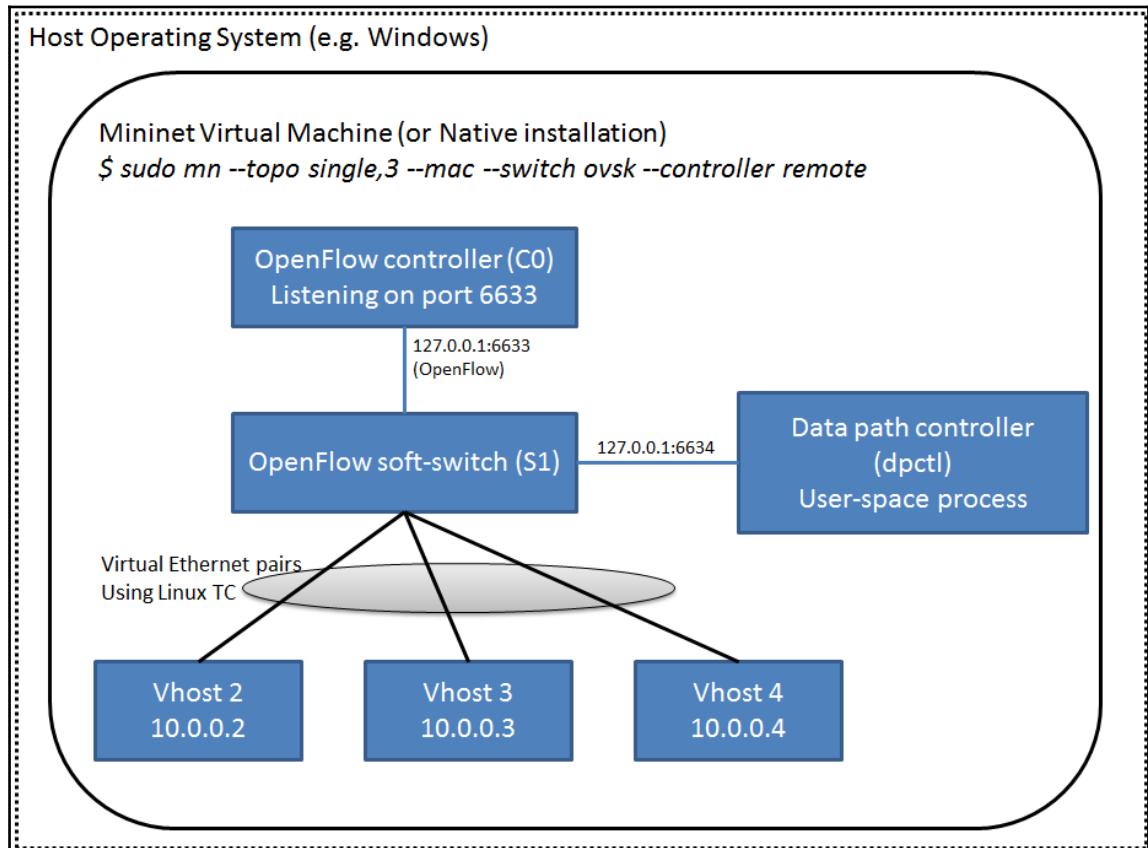


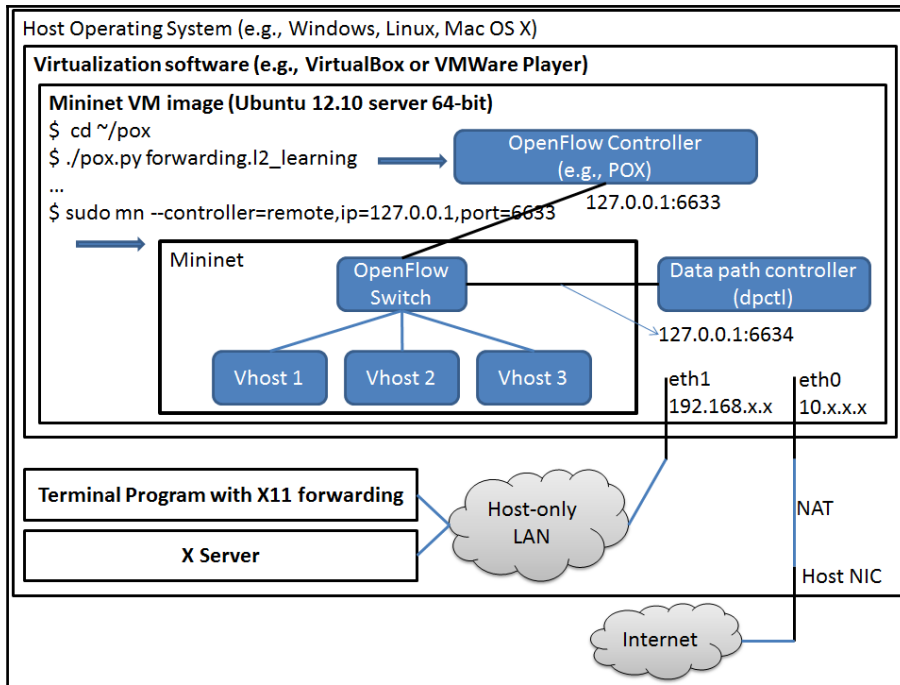
# Chapter 4: The OpenFlow Controllers

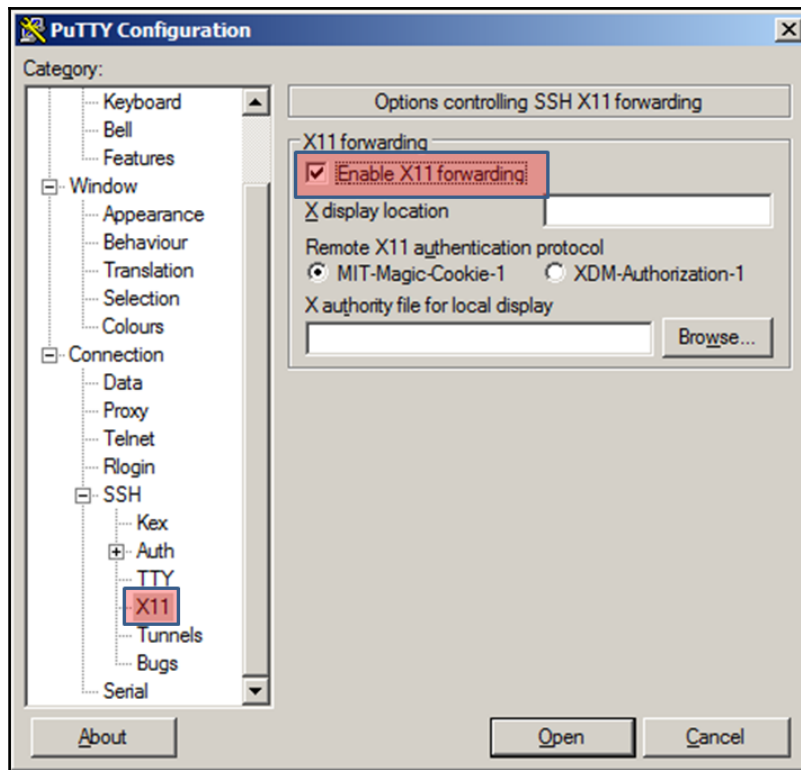




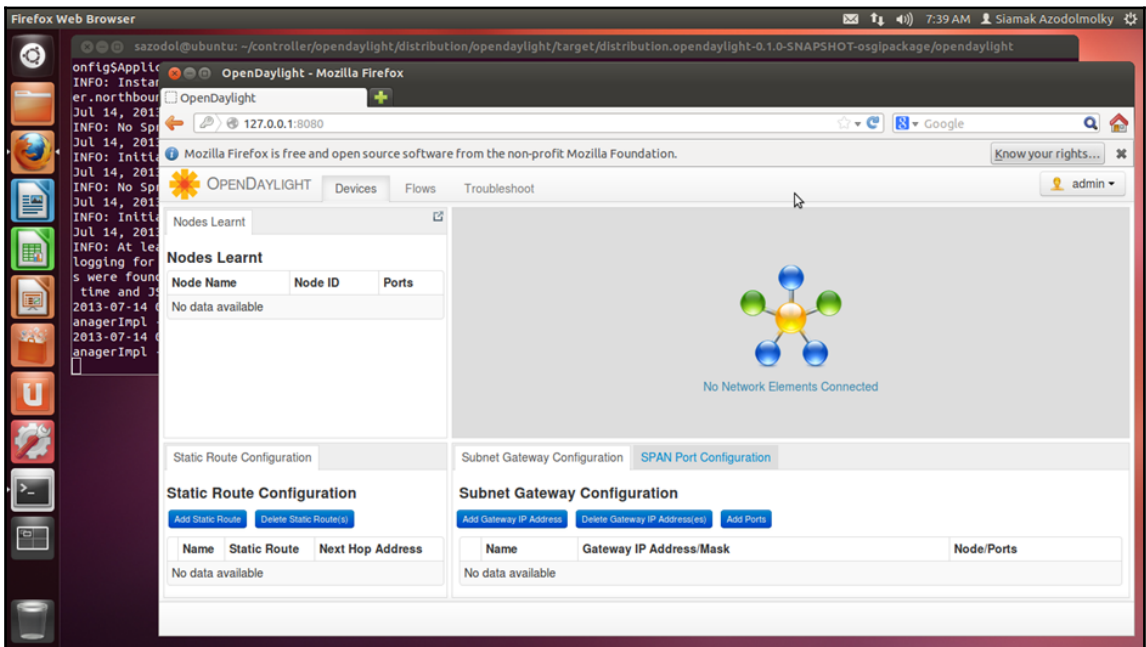
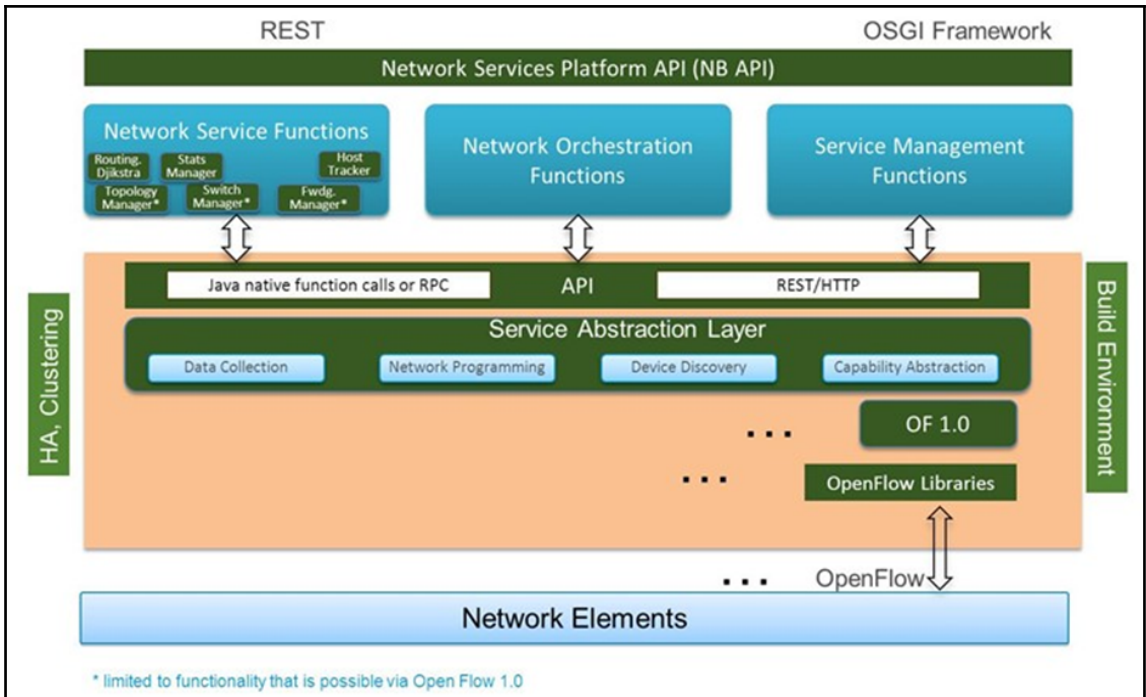
# Chapter 5: Setting Up the Environment

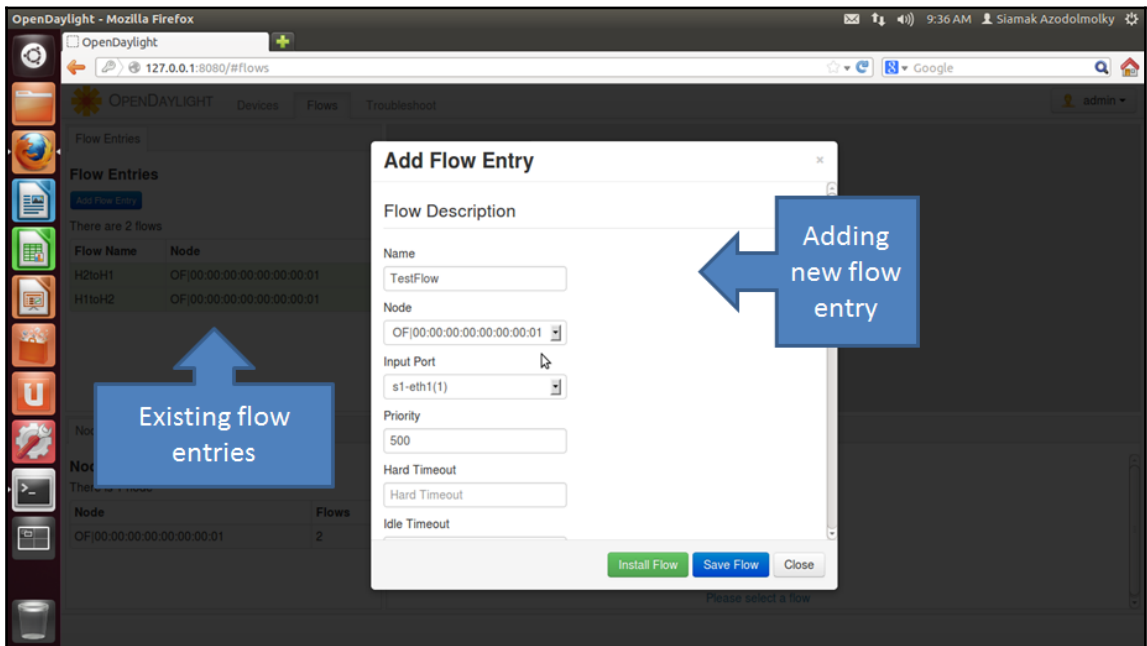
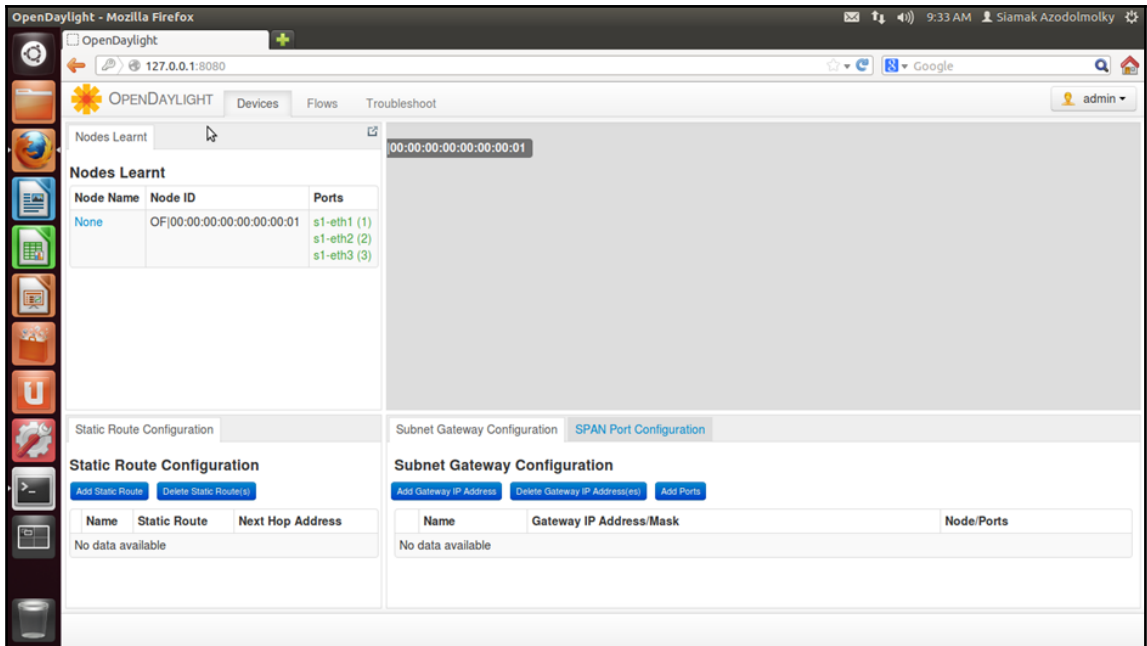




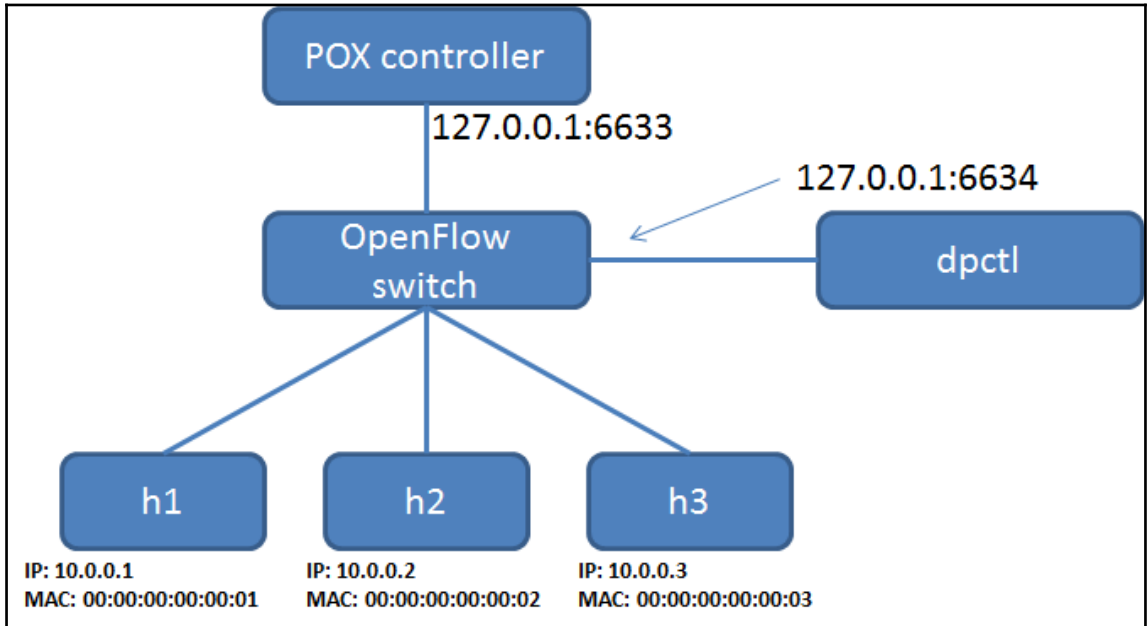








# Chapter 6: Net App Development



OpenDaylight

172.16.102.161:8080

OPENDAYLIGHT Devices Flows Troubleshoot admin

Nodes Learnt

**Nodes Learnt**

Node Name	Node ID	Ports
<a href="#">Click to update</a>	00:00:00:00:00:00:00:07	s7(0) s7-eth1(1) s7-eth2(2) s7-eth3(3)
<a href="#">Click to update</a>	00:00:00:00:00:00:00:06	s6(0) s6-eth1(1) s6-eth2(2) s6-eth3(3)
<a href="#">Click to update</a>	00:00:00:00:00:00:00:05	s5(0) s5-eth1(1) s5-eth2(2) s5-eth3(3)
<a href="#">Click to</a>	00:00:00:00:00:00:00:04	s4(0)

Static route Configuration

[Add Static Route](#) [Delete Static Route\(s\)](#)

Name	Static Route	NextHop address
No data available		

Subnet Gateway Configuration [SPAN Port Configuration](#)

[Add Gateway IP Address](#) [Delete Gateway IP Address\(es\)](#) [Add Ports](#)

Name	Gateway IP Address/Mask	Node/Ports
No data available		

Users Save Logout

172.16.102.161:8080/#save

The screenshot shows the OpenDaylight Troubleshoot interface. The main area displays a network topology with nodes represented by blue cubes. The root node is OF|00:00:00:00:00:00:01, which connects to two intermediate nodes: OF|00:00:00:00:00:00:02 and OF|00:00:00:00:00:00:05. The OF|00:00:00:00:00:00:02 node connects to two leaf nodes: OF|00:00:00:00:00:00:03 and OF|00:00:00:00:00:00:04. The OF|00:00:00:00:00:00:05 node connects to two leaf nodes: OF|00:00:00:00:00:00:07 and OF|00:00:00:00:00:00:08.

On the left, the 'Existing Nodes' table is visible, with the 'Ports' link highlighted in red for the first node:

Node Names	Node ID	Statistics
	OF 00:00:00:00:00:00:07	Flows Ports
	OF 00:00:00:00:00:00:06	Flows Ports
	OF 00:00:00:00:00:00:05	Flows Ports
	OF 00:00:00:00:00:00:04	Flows Ports
	OF 00:00:00:00:00:00:03	Flows Ports

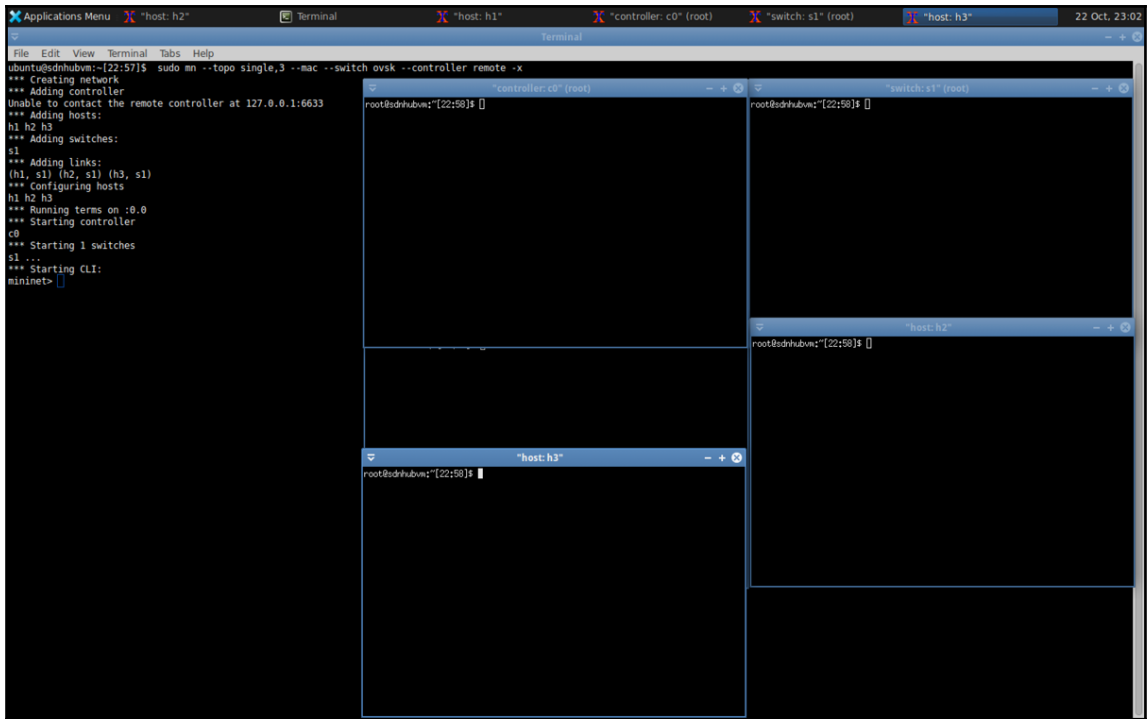
Below the topology, the 'Port Details' table shows statistics for various connectors:

Node Connector	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisic
OF 3@OF 00:00:00:00:00:00:07	87	82	6356	5978	0	0	0	0	0	0	0	0
OF 2@OF 00:00:00:00:00:00:07	5	213	378	15036	0	0	0	0	0	0	0	0
SW@OF 00:00:00:00:00:00:07	0	0	0	0	0	0	0	0	0	0	0	0

```

osgi> ss simple
"Framework is launched."
id State Bundle
45 ACTIVE org.opendaylight.controller.samples.simpleforwarding_0.4.0.SNAPSHOT

```



```
"switch: s1" (root)
root@sdnhubvm:~[23:09]$ ovs-vsctl show
873c293e-912d-4067-82ad-d1116d2ad39f
  Bridge "s1"
    Controller "tcp:127.0.0.1:6633"
    Controller "ptcp:6634"
    fail_mode: secure
    Port "s1-eth3"
      Interface "s1-eth3"
    Port "s1-eth2"
      Interface "s1-eth2"
    Port "s1-eth1"
      Interface "s1-eth1"
    Port "s1"
      Interface "s1"
        type: internal
    ovs_version: "2.3.90"
root@sdnhubvm:~[23:09]$ ovs-vsctl set Bridge s1 protocols=OpenFlow13
root@sdnhubvm:~[23:09]$
```

```
root@sdnhubvm:~[00:52]$ sudo ryu-manager --verbose ryu.app.example_switch_13
/usr/lib/python2.7/dist-packages/pkg_resources.py:1031: UserWarning: /home/ubuntu/.python-eggs is writable by group/others and vulnerable to attack when used with get_resource_filename. Consider a more secure location (set with .set_extraction_path or the PYTHON_EGG_CACHE environment variable).
  warnings.warn(msg, UserWarning)
loading app ryu.app.example_switch_13
loading app ryu.controller.ofp_handler
instantiating app ryu.app.example_switch_13 of ExampleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
BRICK ExampleSwitch13
  CONSUMES EventOFPPacketIn
  CONSUMES EventOFPSwitchFeatures
BRICK ofp_event
  PROVIDES EventOFPPacketIn TO {'ExampleSwitch13': set(['main'])}
  PROVIDES EventOFPSwitchFeatures TO {'ExampleSwitch13': set(['config'])}
  CONSUMES EventOFPPortDescStatsReply
  CONSUMES EventOFPHello
  CONSUMES EventOFPPortStatus
  CONSUMES EventOFPSwitchFeatures
  CONSUMES EventOFPErrormsg
  CONSUMES EventOFPEchoRequest
  CONSUMES EventOFPEchoReply
connected socket:<eventlet.greenio.base.GreenSocket object at 0x7fe695e2fe10> address:('127.0.0.1', 37363)
hello ev <ryu.controller.ofp_event.EventOFPHello object at 0x7fe695dc1290>
move onto config mode
EVENT ofp_event->ExampleSwitch13 EventOFPSwitchFeatures
switch features ev version=0x4,msg_type=0x6,msg_len=0x20,,xid=0x31061b54,OFPSwitchFeatures(auxiliary_id=0,capabilities=79,data_path_id=1,n_buffers=256,n_tables=254)
move onto main mode
```

```
"switch: s1" (root)
root@sdnhubvm:~[01:17]$ ovs-ofctl -O openflow13 dump-flows s1
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=1490.898s, table=0, n_packets=0, n_bytes=0, priority=0 actions=CONTROLLER:65535
root@sdnhubvm:~[01:17]$ █
```

```
"host: h3"
root@sdnhubvm:~[01:58]$ tcpdump -en -i h3-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h3-eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
█
```

```
"switch: s1" (root)
root@sdnhubvm:~[02:12]$ ovs-ofctl -O openflow13 dump-flows s1
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=215.155s, table=0, n_packets=2, n_bytes=140, priority=1,in_port=3,d1_dst=00:00:00:00:00:01 actions=output:1
 cookie=0x0, duration=215.151s, table=0, n_packets=1, n_bytes=42, priority=1,in_port=1,d1_dst=00:00:00:00:00:03 actions=output:3
 cookie=0x0, duration=4793.205s, table=0, n_packets=3, n_bytes=182, priority=0 actions=CONTROLLER:65535
root@sdnhubvm:~[02:12]$ ^C
root@sdnhubvm:~[02:13]$ █
```

```
EVENT ofp_event->ExampleSwitch13 EventOFPPacketIn
packet in 1 00:00:00:00:00:01 ff:ff:ff:ff:ff:ff 1
EVENT ofp_event->ExampleSwitch13 EventOFPPacketIn
packet in 1 00:00:00:00:00:03 00:00:00:00:00:01 3
EVENT ofp_event->ExampleSwitch13 EventOFPPacketIn
packet in 1 00:00:00:00:00:01 00:00:00:00:00:03 1
█
```

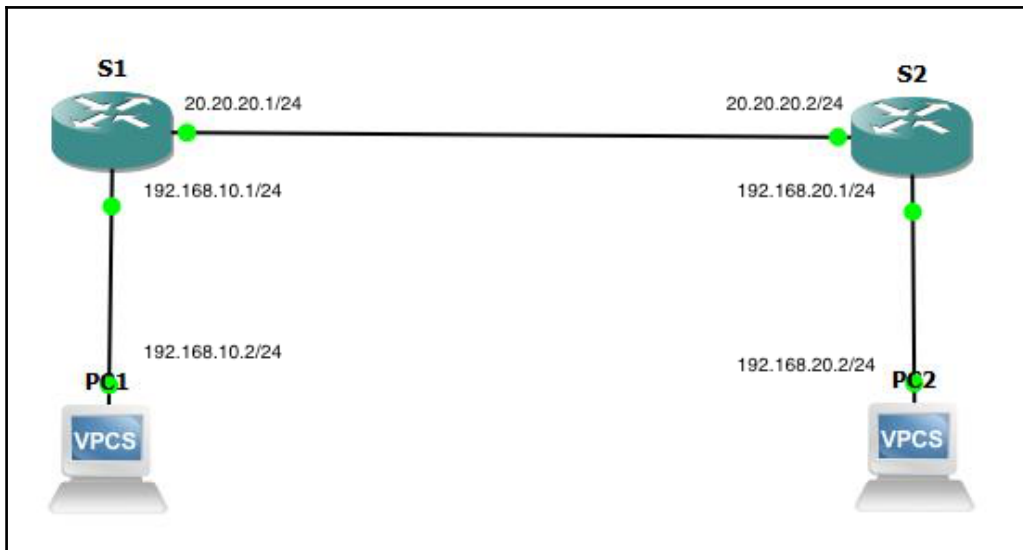
```
"host: h1"
root@sdnhubvm:~[00:50]$ tcpdump -en -i h1-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h1-eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
02:08:58.978415 00:00:00:00:00:01 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), 1
length 42: Request who-has 10.0.0.3 tell 10.0.0.1, length 28
02:08:58.984576 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype ARP (0x0806), 1
length 42: Reply 10.0.0.3 is-at 00:00:00:00:00:03, length 28
02:08:58.984614 00:00:00:00:00:01 > 00:00:00:00:00:03, ethertype IPv4 (0x0800),
length 98: 10.0.0.1 > 10.0.0.3: ICMP echo request, id 23795, seq 1, length 64
02:08:58.988790 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype IPv4 (0x0800),
length 98: 10.0.0.3 > 10.0.0.1: ICMP echo reply, id 23795, seq 1, length 64
02:09:04.003208 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype ARP (0x0806), 1
length 42: Request who-has 10.0.0.1 tell 10.0.0.3, length 28
02:09:04.003224 00:00:00:00:00:01 > 00:00:00:00:00:03, ethertype ARP (0x0806), 1
length 42: Reply 10.0.0.1 is-at 00:00:00:00:00:01, length 28
```



```

root@sdnhubvm:~# [01:58]$ tcpdump -en -i h3-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h3-eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
02:08:58.980664 00:00:00:00:00:01 > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 42: Request who-has 10.0.0.3 tell 10.0.0.1, length 28
02:08:58.980689 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype ARP (0x0806), length 42: Reply 10.0.0.3 is-at 00:00:00:00:00:03, length 28
02:08:58.988326 00:00:00:00:00:01 > 00:00:00:00:00:03, ethertype IPv4 (0x0800), length 98: 10.0.0.1 > 10.0.0.3: ICMP echo request, id 23795, seq 1, length 64
02:08:58.988380 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype IPv4 (0x0800), length 98: 10.0.0.3 > 10.0.0.1: ICMP echo reply, id 23795, seq 1, length 64
02:09:04.003036 00:00:00:00:00:03 > 00:00:00:00:00:01, ethertype ARP (0x0806), length 42: Request who-has 10.0.0.1 tell 10.0.0.3, length 28
02:09:04.003329 00:00:00:00:00:01 > 00:00:00:00:00:03, ethertype ARP (0x0806), length 42: Reply 10.0.0.1 is-at 00:00:00:00:00:01, length 28

```



```

ubuntu@sdnhubvm:~# [06:32]$ sudo mn --topo linear,2 --mac --switch ovsk,protocols=OpenFlow13 --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s2) (s2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet>

```

```
mininet> h1 route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.10.1   0.0.0.0         UG    0      0      0 h1-eth0
192.168.10.0     0.0.0.0        255.255.255.0   U     0      0      0 h1-eth0
```

```
mininet> h2 route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.20.1   0.0.0.0         UG    0      0      0 h2-eth0
192.168.20.0     0.0.0.0        255.255.255.0   U     0      0      0 h2-eth0
```

```
ubuntu@sdnhubvm:~[00:50]$ cd ryu
ubuntu@sdnhubvm:~/ryu[07:05] (master)$ bin/ryu-manager --verbose ryu/app/rest_router.py
loading app ryu/app/rest_router.py
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
creating context wsgi
instantiating app ryu/app/rest_router.py of RestRouterAPI
instantiating app ryu.controller.ofp_handler of OFPHandler
BRICK dpset
  PROVIDES EventDP TO {'RestRouterAPI': set(['dpset'])}
  CONSUMES EventOFPSwitchFeatures
  CONSUMES EventOFPPortStatus
  CONSUMES EventOFPPStateChange
BRICK RestRouterAPI
  CONSUMES EventDP
  CONSUMES EventOFPPFlowStatsReply
  CONSUMES EventOFPPStatsReply
  CONSUMES EventOFPPacketIn
BRICK ofp_event
  PROVIDES EventOFPPacketIn TO {'RestRouterAPI': set(['main'])}
  PROVIDES EventOFPPStatsReply TO {'RestRouterAPI': set(['main'])}
  PROVIDES EventOFPPortStatus TO {'dpset': set(['main'])}
  PROVIDES EventOFPPStateChange TO {'dpset': set(['main', 'dead'])}
  PROVIDES EventOFPSwitchFeatures TO {'dpset': set(['config'])}
  PROVIDES EventOFPPFlowStatsReply TO {'RestRouterAPI': set(['main'])}
  CONSUMES EventOFPHello
  CONSUMES EventOFPErrormsg
  CONSUMES EventOFPEchoRequest
  CONSUMES EventOFPEchoReply
  CONSUMES EventOFPPortDescStatsReply
  CONSUMES EventOFPSwitchFeatures
  CONSUMES EventOFPPortStatus
(24727) wsgi starting up on http://0.0.0.0:8080/
```

```

[RT][INFO] switch_id=0000000000000001: Set SW config for TTL error packet in.
[RT][INFO] switch_id=0000000000000001: Set ARP handling (packet in) flow [cookie=0x0]
[RT][INFO] switch_id=0000000000000001: Set L2 switching (normal) flow [cookie=0x0]
[RT][INFO] switch_id=0000000000000001: Set default route (drop) flow [cookie=0x0]
[RT][INFO] switch_id=0000000000000001: Start cyclic routing table update.
[RT][INFO] switch_id=0000000000000001: Join as router.
[RT][INFO] switch_id=0000000000000002: Set SW config for TTL error packet in.
[RT][INFO] switch_id=0000000000000002: Set ARP handling (packet in) flow [cookie=0x0]
[RT][INFO] switch_id=0000000000000002: Set L2 switching (normal) flow [cookie=0x0]
[RT][INFO] switch_id=0000000000000002: Set default route (drop) flow [cookie=0x0]
[RT][INFO] switch_id=0000000000000002: Start cyclic routing table update.
[RT][INFO] switch_id=0000000000000002: Join as router.

```

```

"Node: c0" (root)
</body>
root@sdnhubvm:~[07:51]$ curl -X POST -d '{"address":"192.168.10.1/24"}' http://localhost:8080/router/0000000000000001
[{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "Add address [address_id=1]"}]}root@sdnhubvm:~[07:54]$
root@sdnhubvm:~[07:54]$
root@sdnhubvm:~[07:56]$
root@sdnhubvm:~[07:56]$

```

```

"Node: c0" (root)
root@sdnhubvm:~[08:36]$ curl -X POST -d '{"address":"20.20.20.1/24"}' http://localhost:8080/router/0000000000000001
[{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "Add address [address_id=2]"}]}root@sdnhubvm:~[08:36]$
root@sdnhubvm:~[08:36]$

```

```

"Node: c0" (root)
root@sdnhubvm:~[07:56]$ curl -X POST -d '{"address":"192.168.20.1/24"}' http://localhost:8080/router/0000000000000002
[{"switch_id": "0000000000000002", "command_result": [{"result": "success", "details": "Add address [address_id=1]"}]}root@sdnhubvm:~[07:58]$
root@sdnhubvm:~[07:58]$
root@sdnhubvm:~[07:58]$
root@sdnhubvm:~[07:58]$

```

```

"Node: c0" (root)
root@sdnhubvm:~[08:36]$ curl -X POST -d '{"address":"20.20.20.2/24"}' http://localhost:8080/router/0000000000000002
[{"switch_id": "0000000000000002", "command_result": [{"result": "success", "details": "Add address [address_id=2]"}]}root@sdnhubvm:~[08:37]$
root@sdnhubvm:~[08:37]$
root@sdnhubvm:~[08:37]$

```

```

"Node: c0" (root)
root@sdnhubvm:~[08:40]$ curl -X POST -d '{"gateway":"20.20.20.2"}' http://localhost:8080/router/0000000000000001
[{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "Add route [route_id=1]"}]}root@sdnhubvm:~[08:41]$
root@sdnhubvm:~[08:41]$

```

```

"Node: c0" (root)
</html>root@sdnhubvm:~[08:39]$ curl -X POST -d '{"gateway":"20.20.20.1"}' http://localhost:8080/router/0000000000000002
[{"switch_id": "0000000000000002", "command_result": [{"result": "success", "details": "Add route [route_id=1]"}]}root@sdnhubvm:~[08:39]$
root@sdnhubvm:~[08:40]$
root@sdnhubvm:~[08:40]$

```

```

mininet> h1 ping h2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
From 192.168.10.2 icmp_seq=1 Destination Host Unreachable
From 192.168.10.2 icmp_seq=2 Destination Host Unreachable
From 192.168.10.2 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.20.2 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4000ms
pipe 3
mininet> xterm c0
mininet> xterm c0
mininet> h1 ping h2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=62 time=14.8 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=62 time=0.292 ms
64 bytes from 192.168.20.2: icmp_seq=3 ttl=62 time=0.060 ms
64 bytes from 192.168.20.2: icmp_seq=4 ttl=62 time=0.062 ms
64 bytes from 192.168.20.2: icmp_seq=5 ttl=62 time=0.078 ms
^C
--- 192.168.20.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 0.060/3.076/14.891/5.908 ms
mininet>

```

```

ubuntu@sdnhubvm:~[13:29]$ sudo mn --topo single,3 --mac --switch ovsk,protocols=OpenFlow13 --controller remote -x
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Running terms on :0.0
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
mininet>

```

```
[FW][INFO] dpid=0000000000000001: Join as firewall.
```

```

"controller: c0" (root)
root@sdnhubvm:~[14:28]$ curl -X PUT http://localhost:8080/firewall/module/enable/0000000000000001
[{"switch_id": "0000000000000001", "command_result": {"result": "success", "details": "firewall running."}}]root@sdnhubvm:~[14:30]$
root@sdnhubvm:~[14:30]$
root@sdnhubvm:~[14:30]$

```

```
"controller: c0" (root)
root@sdnhubvm:~# curl http://localhost:8080/firewall/module/status
[{"status": "enable", "switch_id": "0000000000000001"}]root@sdnhubvm:~#
```

```
"controller: c0" (root)
root@sdnhubvm:~# curl -X POST -d '{"nw_src": "10.0.0.1/32", "nw_dst": "10.0.0.2/32", "nw_proto": "ICMP"}' http://localhost:8080/firewall/rules/0000000000000001
[{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "Rule added. ; rule_id=2"}]]root@sdnhubvm:~#
```

```
"controller: c0" (root)
root@sdnhubvm:~# curl http://localhost:8080/firewall/rules/0000000000000001
[{"access_control_list": [{"rules": [{"priority": 1, "dl_type": "ARP", "rule_id": 0, "actions": "DENY"}, {"priority": 1, "dl_type": "IPv4", "rule_id": 0, "actions": "DENY"}, {"priority": 1, "dl_type": "IPv4", "nw_proto": "ICMP", "nw_dst": "10.0.0.2", "nw_src": "10.0.0.1", "rule_id": 2, "actions": "ALLOW"}, {"priority": 1, "dl_type": "IPv4", "nw_proto": "ICMP", "nw_dst": "10.0.0.1", "nw_src": "10.0.0.2", "rule_id": 3, "actions": "ALLOW"}]}], "switch_id": "0000000000000001"}]root@sdnhubvm:~#
```

```
"host: h2"
root@sdnhubvm:~# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.236 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.106 ms
^C
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.051/0.131/0.236/0.077 ms
root@sdnhubvm:~# ping 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.

^C
--- 10.0.0.3 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2010ms

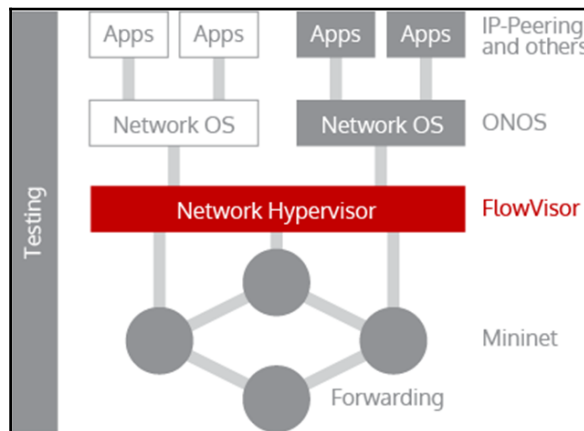
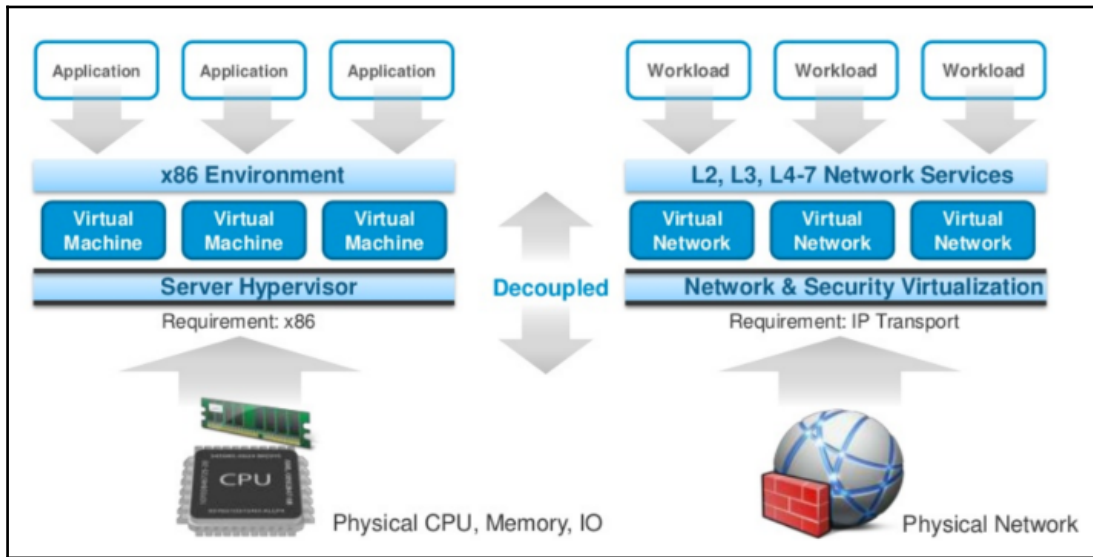
root@sdnhubvm:~#
```

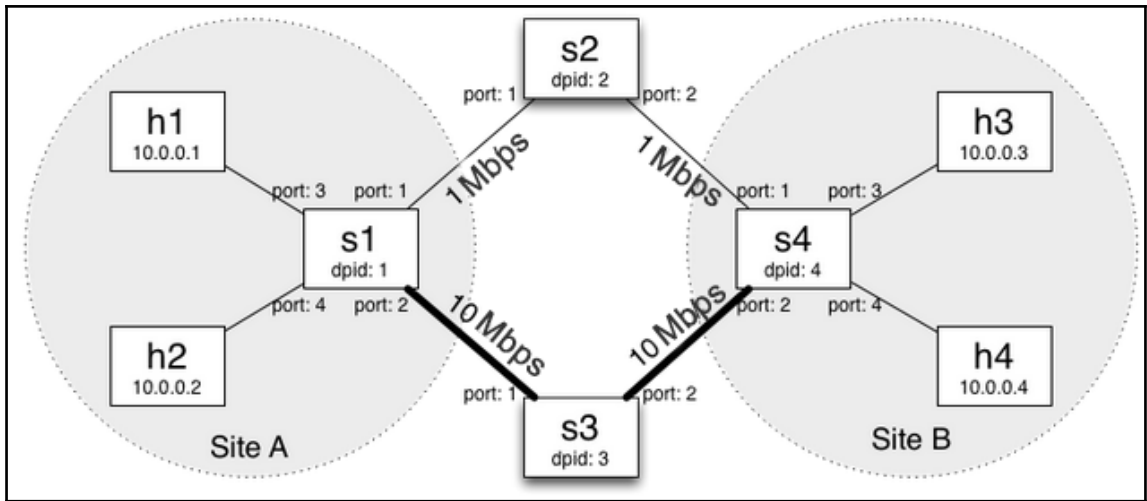
```
host: h1
root@sdnhubvm:~[15:10]$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.34 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.126 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 0.059/0.510/1.345/0.591 ms
root@sdnhubvm:~[15:13]$ ping 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.

^C
--- 10.0.0.3 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3003ms

root@sdnhubvm:~[15:13]$
```

# Chapter 7: Getting a Network Slice



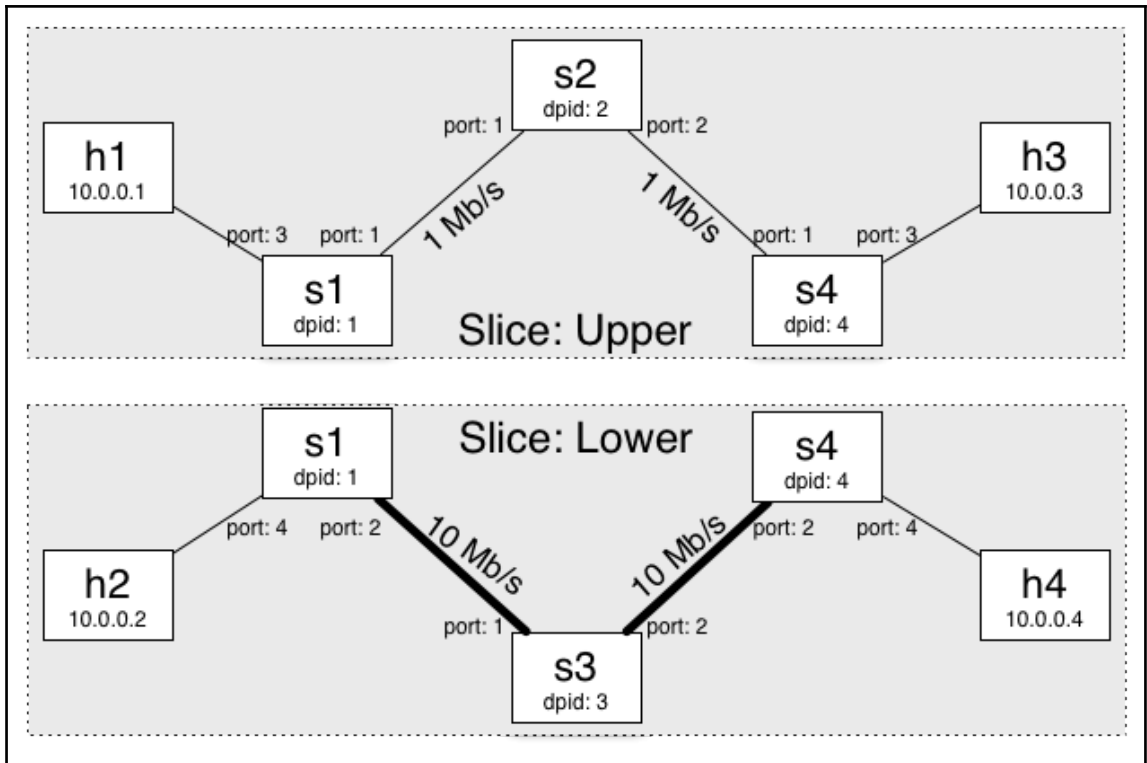


```

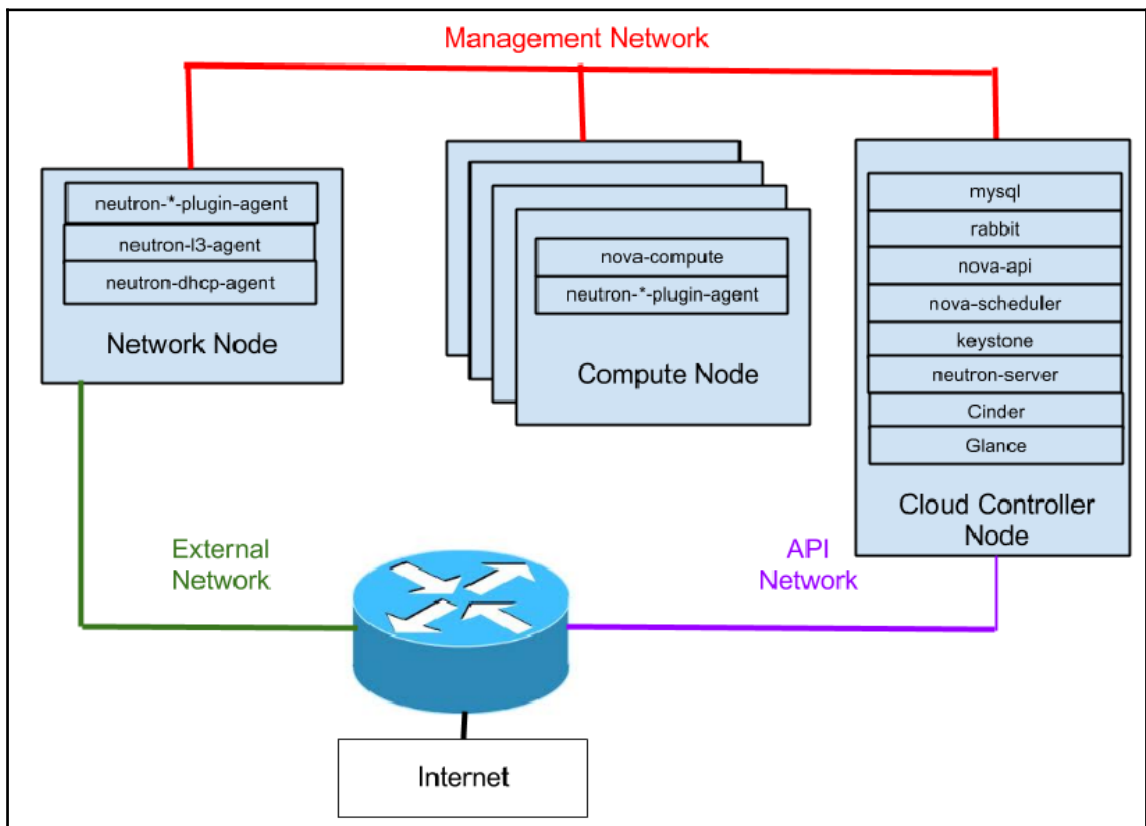
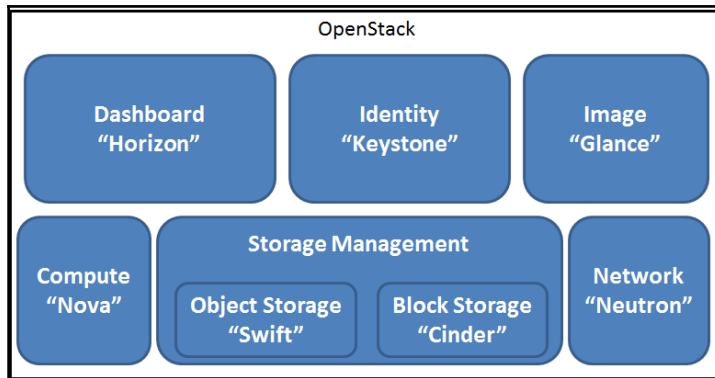
{
  "enable-topo-ctrl": true ,
  "flood-perm": {
    "dpid": "all",
    "slice-name": "fvadmin"
  },
  "flow-stats-cache": 30,
  "flowmod-limit": {
    "fvadmin": {
      "00:00:00:00:00:00:00:01": -1,
      "00:00:00:00:00:00:00:02": -1,
      "00:00:00:00:00:00:00:03": -1,
      "00:00:00:00:00:00:00:04": -1,
      "any": null
    }
  },
  "stats-desc": false,
  "track-flows": false
}

```



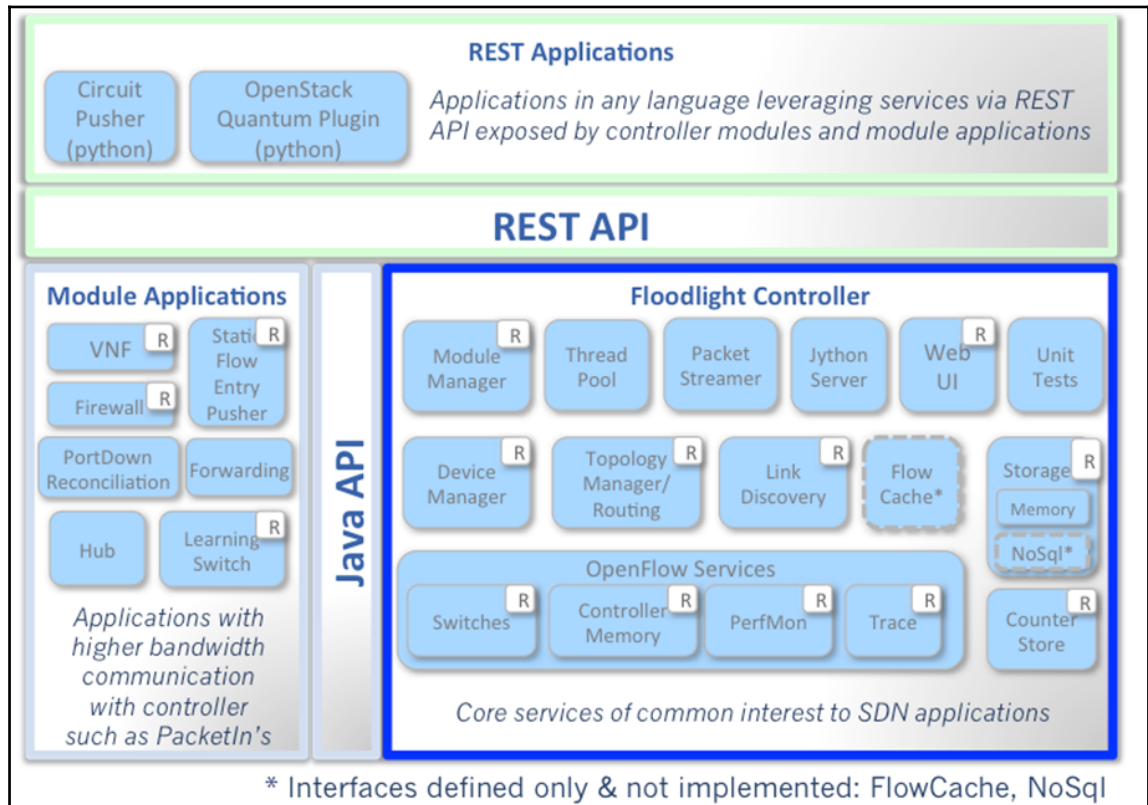


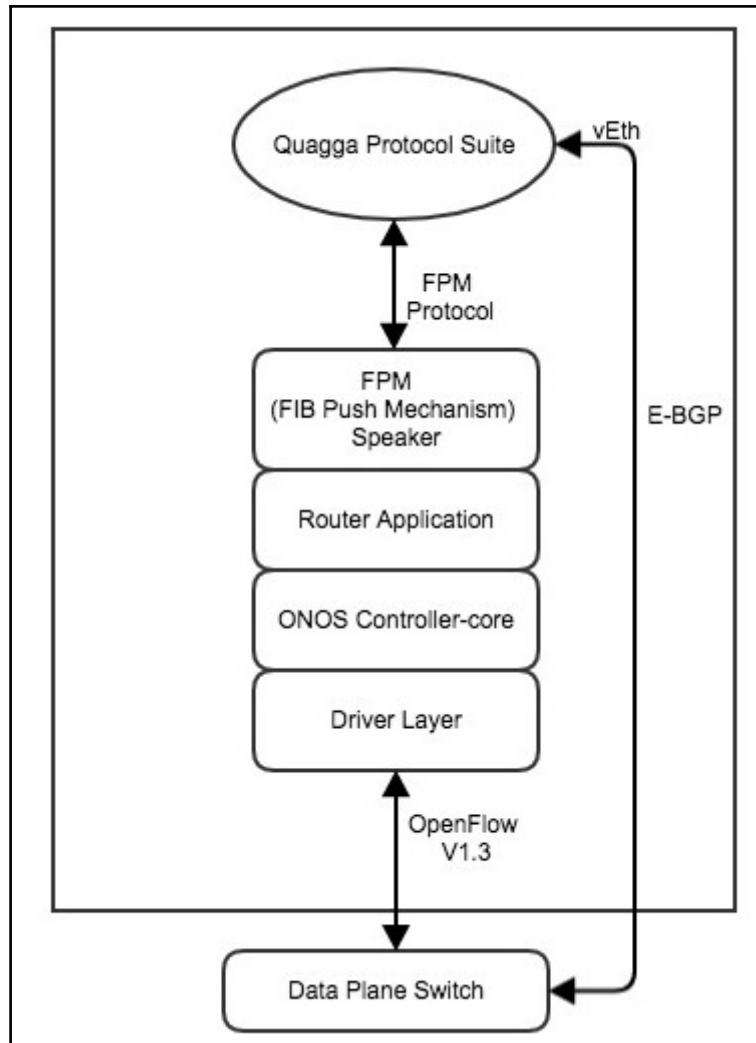
# Chapter 8: OpenFlow in Cloud Computing

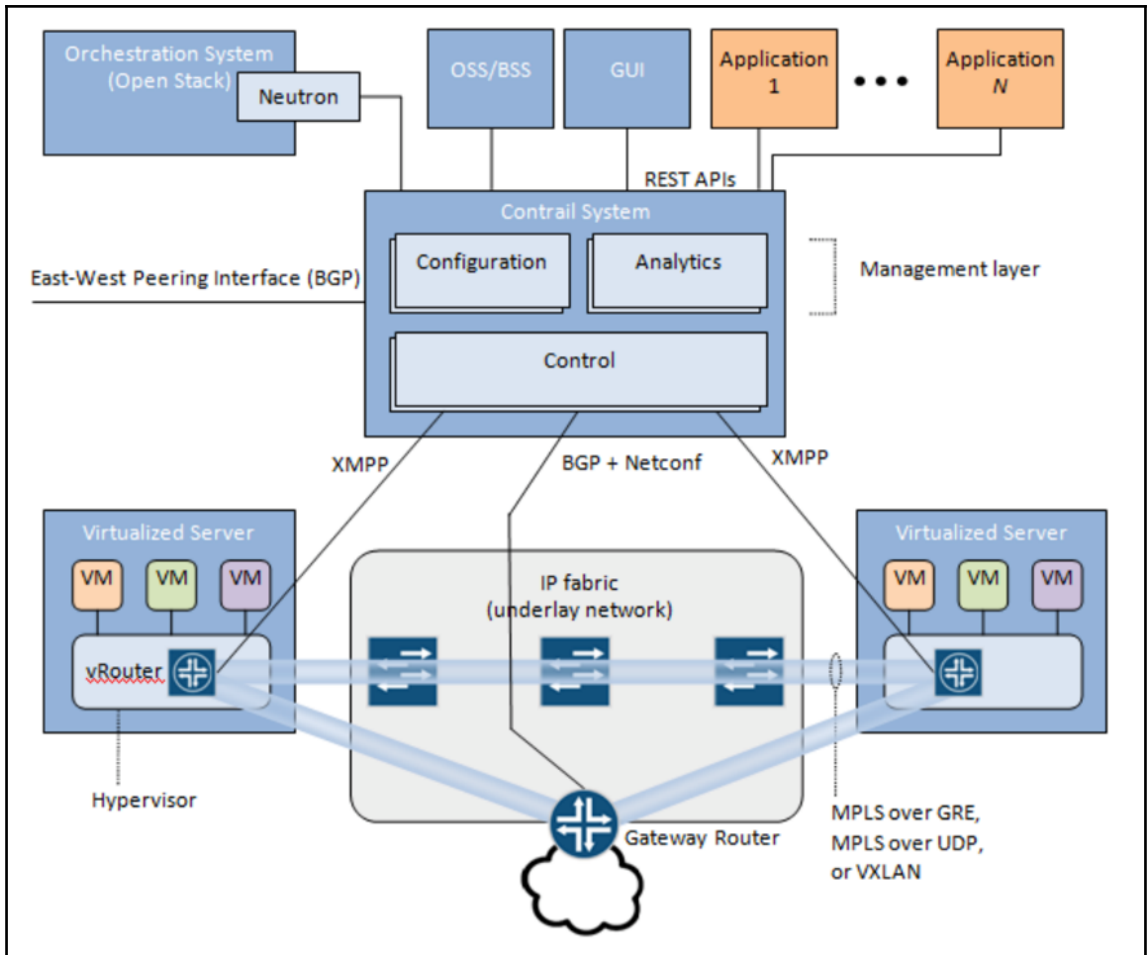


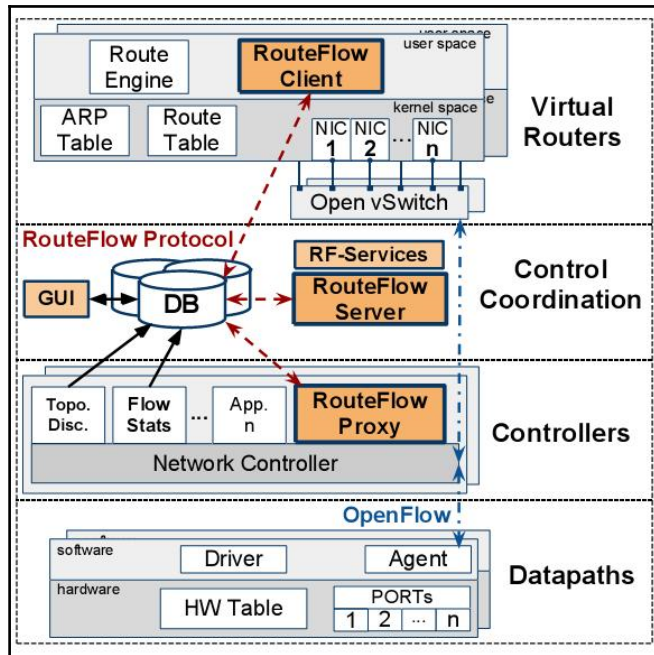
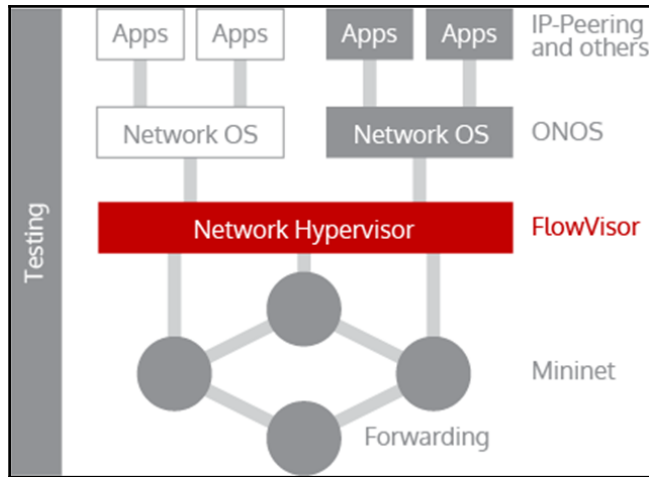
```
Horizon is now available at http://10.10.2.15 /  
Keystone is serving at http://10.10.2.15:5000/v2.0/  
Examples on using novaclient command line is in  
exercise.sh  
The default users are: admin and demo  
The password: nova  
This is your host ip: 10.10.2.15  
stack.sh completed in 103 seconds.
```

# Chapter 9: Open Source Resources

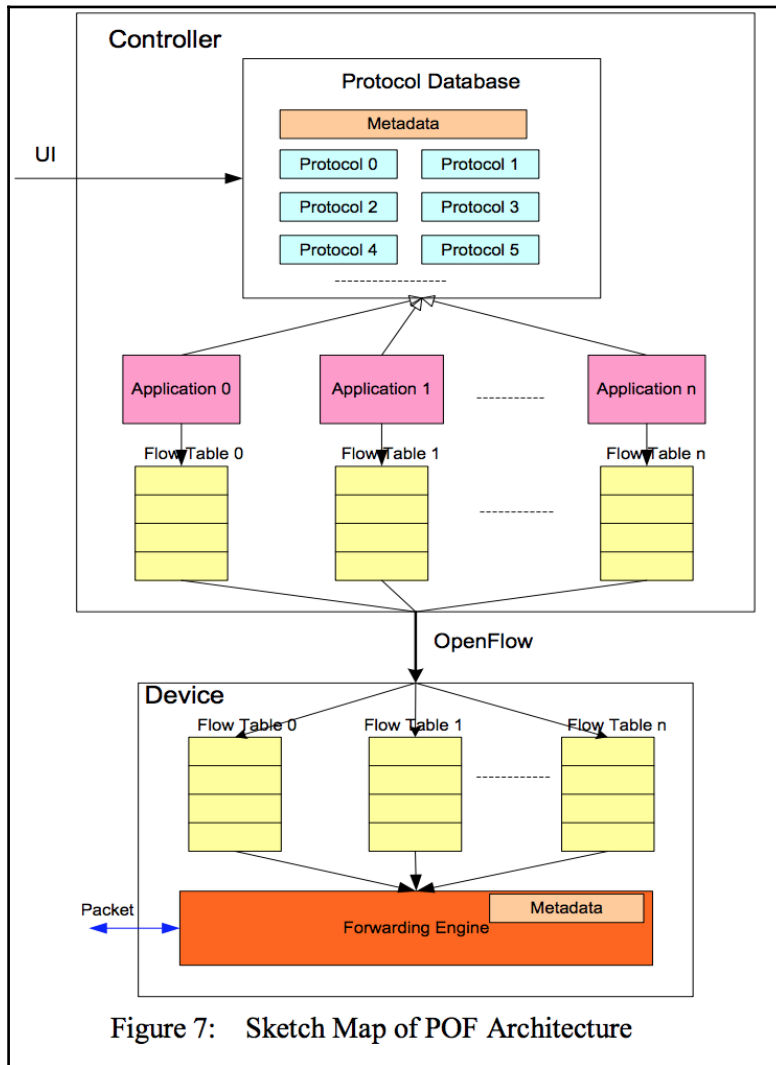




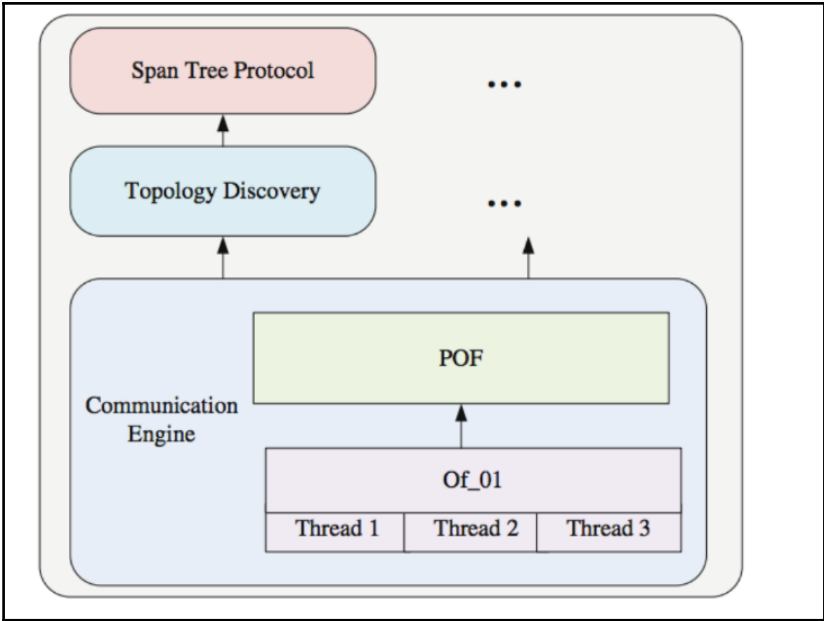


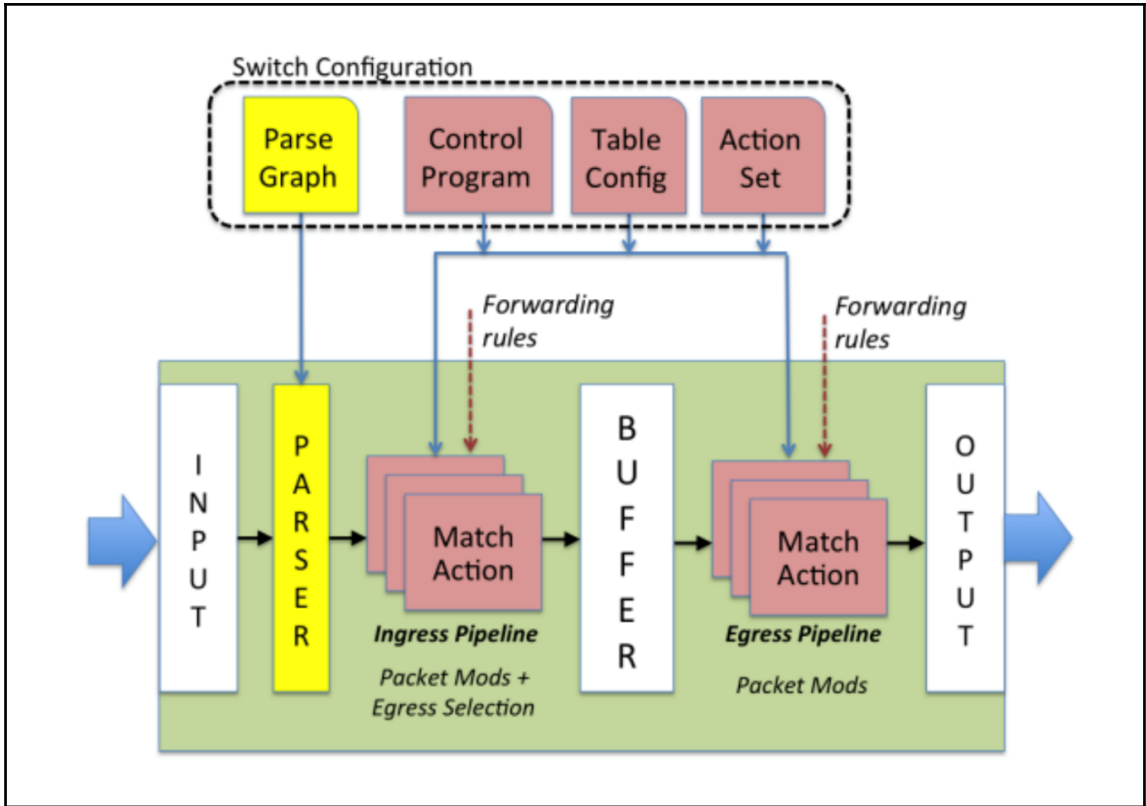


# Chapter 10: The Future of SDN









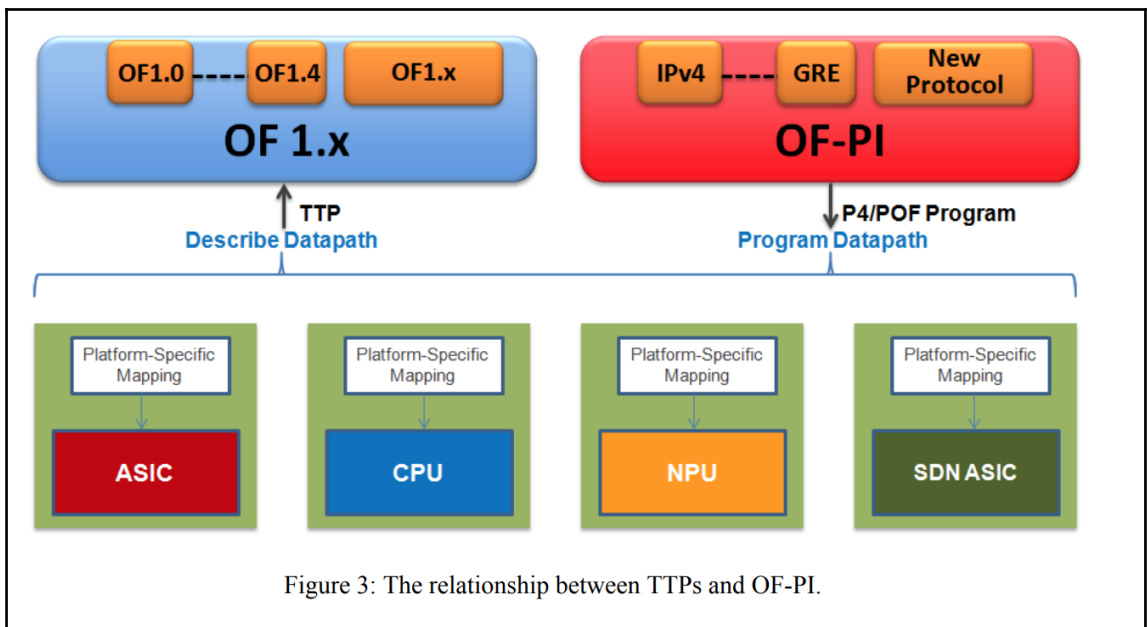
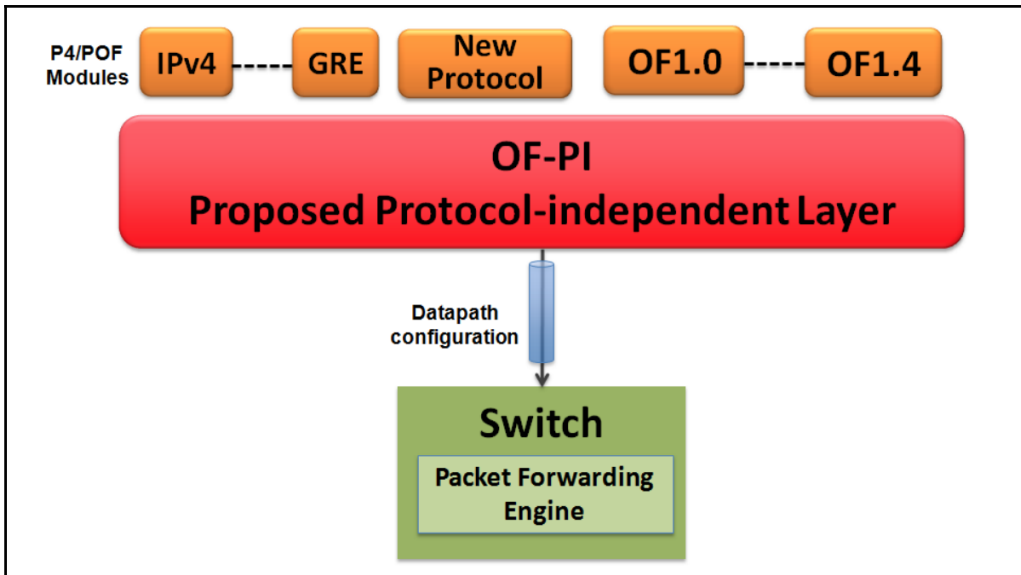


Figure 3: The relationship between TTPs and OF-PI.

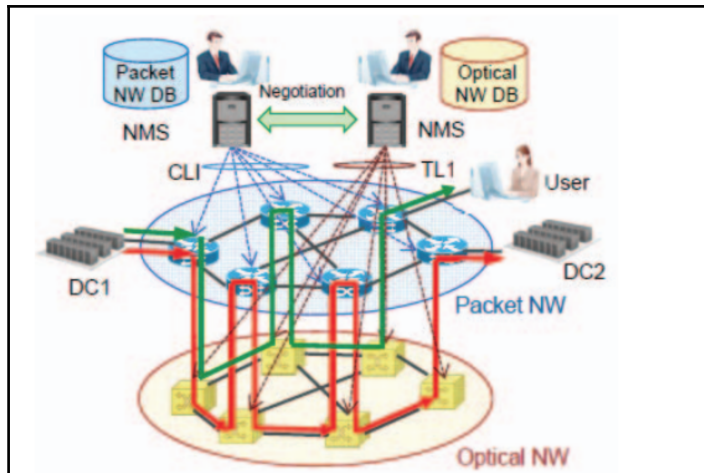


Fig.1. Current operation system for multilayer networks

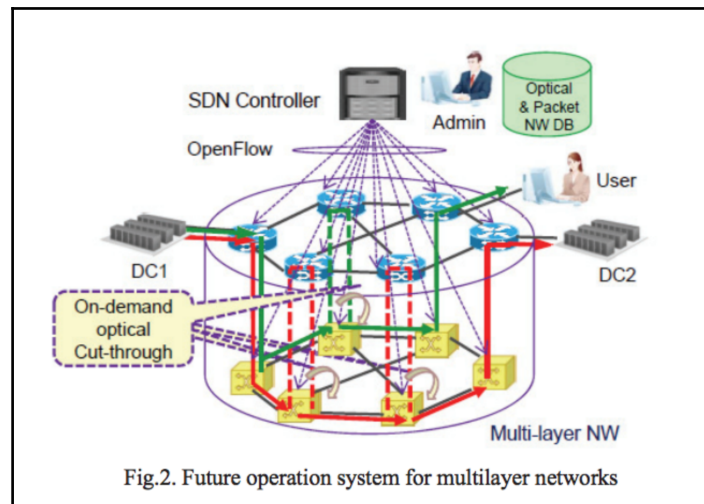


Fig.2. Future operation system for multilayer networks

