Example 2:

h1(192.168.0.2)--wireless link--h0 (wireless router:192.168.0.1 and 10.0.01)-- s0(OVS)--h3(10.0.0.2)

h2 (192.168.0.3)--wireless link—

Usage:
1. Open the vm.
2. Login with name: mininet and password: mininet
3. Change to graphical mode: startx
4. Open a terminal
5. Go to the directory containing ns3: cd ns-allinone-3.17/ns-3.17
6. Run sudo ./waf shell in order to let the ns3 set appropriate environment variables.
7. Run the test-wifi2.py

```
#!/usr/bin/python

"""
This example shows how to create an empty Mininet object
(without a topology object) and add nodes to it manually.
"""

from mininet.net import Mininet
from mininet.node import OVSController
from mininet.cli import CLI
from mininet.log import setLogLevel, info

import mininet.ns3                      # line added
from mininet.ns3 import WIFISegment
from mininet.link import TCLink
import ns.wifi

def emptyNet():

    net = Mininet( controller=OVSController )

    info( '*** Adding controller\n' )
    net.addController( 'c0' )

    info( '*** Adding switch\n' )
    s0 = net.addSwitch( 's0' )
    s0.listenPort = 6634

    h0 = net.addHost( 'h0' )
    h1 = net.addHost( 'h1', ip='192.168.0.2' )
    h2 = net.addHost( 'h2', ip='192.168.0.3' )
    h3 = net.addHost( 'h3', ip='10.0.0.2' )
    linkopts=dict( bw=100, delay='1ms', loss=0 )
    TCLink( s0, h3, **linkopts )
    TCLink( h0, s0, **linkopts )

    wifi = WIFISegment()

    wifi.addAp( h0 )
    wifi.addSta( h1 )
    wifi.addSta( h2 )

    info( '*** Starting network\n')
    net.start()
    mininet.ns3.start()

    h0.cmdPrint ( "ifconfig h0-eth0 10.0.0.1 netmask 255.255.255.0" )
    h0.cmdPrint ( "ifconfig h0-eth1 192.168.0.1 netmask 255.255.255.0" )
    h0.cmdPrint ( "sudo echo 1 > /proc/sys/net/ipv4/ip_forward" )
    h1.cmdPrint ( "route add default gw 192.168.0.1" )
    h2.cmdPrint ( "route add default gw 192.168.0.1" )
    h3.cmdPrint ( "route add default gw 10.0.0.1" )

    info( '*** Testing network connectivity\n' )
    net.pingAll()

    info( '*** Running CLI\n' )
    CLI( net )

    info( '*** Stopping network' )
    mininet.ns3.stop()
    mininet.ns3.clear()                 # line added
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()
```

```
root@mininet-vm:~/mininet/examples/ns3# python test-wifi2.py
*** Adding controller
*** Adding switch
(100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss) *** Starting network
*** Configuring hosts
h0 h1 h2 h3
*** Starting controller
*** Starting 1 switches
s0 (100.00Mbit 1ms delay 0% loss) (100.00Mbit 1ms delay 0% loss)
*** h0 : ('ifconfig h0-eth0 10.0.0.1 netmask 255.255.255.0',)
*** h0 : ('ifconfig h0-eth1 192.168.0.1 netmask 255.255.255.0',)
*** h0 : ('sudo echo 1 > /proc/sys/net/ipv4/ip_forward',)
*** h1 : ('route add default gw 192.168.0.1',)
*** h2 : ('route add default gw 192.168.0.1',)
*** h3 : ('route add default gw 10.0.0.1',)
*** Testing network connectivity
*** Ping: testing ping reachability
h0 -> h1 h2 h3
h1 -> h0 h2 h3
h2 -> h0 h1 h3
h3 -> h0 h1 h2
*** Results: 0% dropped (0/12 lost)
*** Running CLI
*** Starting CLI:
mininet> h1 ping -c 1 h3
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=63 time=5.63 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.638/5.638/5.638/0.000 ms
mininet> h2 ping -c 1 h3
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=63 time=5.79 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.794/5.794/5.794/0.000 ms
```

8. Show the network topology.

```
mininet> net
c0
h0 lo:  h0-eth1:h3-eth0 s0-eth2:h0-eth0
h0 h0-eth0:s0-eth2 h0-eth1:
h1 h1-eth0:
h2 h2-eth0:
h3 h3-eth0:s0-eth1
mininet> dump
<OVSController c0: 127.0.0.1:6633 pid=3203>
<OVSSwitch s0: lo:127.0.0.1,s0-eth1:None,s0-eth2:None pid=3211>
<Host h0: h0-eth0:10.0.0.1,h0-eth1:None pid=3218>
<Host h1: h1-eth0:192.168.0.2 pid=3219>
<Host h2: h2-eth0:192.168.0.3 pid=3220>
<Host h3: h3-eth0:10.0.0.2 pid=3221>
mininet>
```

9. Use the dpctl to show switch s0

```
mininet> s0 dpctl show tcp:127.0.0.1:6634
features_reply (xid=0x72b8f801): ver:0x1, dpid:a250241ac44a
n_tables:255, n_buffers:256
features: capabilities:0xc7, actions:0xfff
 1(s0-eth1): addr:ee:46:b7:73:00:b9, config: 0, state:0
     current:    10GB-FD COPPER
 2(s0-eth2): addr:42:6a:6d:c4:f5:25, config: 0, state:0
     current:    10GB-FD COPPER
 LOCAL(s0): addr:a2:50:24:1a:dc:4a, config: 0x1, state:0x1
get_config_reply (xid=0xaaef391c): miss_send_len=0
mininet>
```

10. Use the dpctl to dump-flows for switch s0

```
mininet> s0 dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0xa728df6): flags=none type=1(flow)
 cookie=0, duration_sec=25s, duration_nsec=420000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=2,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=da:4e:dc:57:63:9b,dl_dst=06:89:c0:40:58:73,nw_src=192.168.0.2,nw_dst=10.0.0.2,nw_tos=0x00,icmp_type=0,icmp_code=0,actions=output:1
 cookie=0, duration_sec=25s, duration_nsec=467000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=1,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=06:89:c0:40:58:73,dl_dst=da:4e:dc:57:63:9b,nw_src=10.0.0.2,nw_dst=192.168.0.2,nw_tos=0x00,icmp_type=0,icmp_code=0,actions=output:2
 cookie=0, duration_sec=25s, duration_nsec=472000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=2,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=da:4e:dc:57:63:9b,dl_dst=06:89:c0:40:58:73,nw_src=192.168.0.2,nw_dst=10.0.0.2,nw_tos=0x00,icmp_type=8,icmp_code=0,actions=output:1
 cookie=0, duration_sec=25s, duration_nsec=424000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=1,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=06:89:c0:40:58:73,dl_dst=da:4e:dc:57:63:9b,nw_src=10.0.0.2,nw_dst=192.168.0.2,nw_tos=0x00,icmp_type=8,icmp_code=0,actions=output:2
 cookie=0, duration_sec=25s, duration_nsec=437000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=2,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=da:4e:dc:57:63:9b,dl_dst=06:89:c0:40:58:73,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0x00,icmp_type=8,icmp_code=0,actions=output:1
 cookie=0, duration_sec=25s, duration_nsec=418000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=1,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=da:4e:dc:57:63:9b,dl_dst=06:89:c0:40:58:73,nw_src=192.168.0.1,nw_dst=10.0.0.2,nw_tos=0x00,icmp_type=8,icmp_code=0,actions=output:2
 cookie=0, duration_sec=25s, duration_nsec=404000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=2,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=da:4e:dc:57:63:9b,dl_dst=06:89:c0:40:58:73,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0x00,icmp_type=8,icmp_code=0,actions=output:1
 cookie=0, duration_sec=25s, duration_nsec=442000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=1,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=06:89:c0:40:58:73,dl_dst=da:4e:dc:57:63:9b,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0x00,icmp_type=0,icmp_code=0,actions=output:2
 cookie=0, duration_sec=25s, duration_nsec=447000000s, table_id=0, priority=65535, n_packets=1, n_bytes=98, idle_timeout=60,hard_timeout=0,icmp,in_port=2,dl_vlan=0xffff,dl_vlan_pcp=0x00,dl_src=da:4e:dc:57:63:9b,dl_dst=06:89:c0:40:58:73,nw_src=192.168.0.1,nw_dst=10.0.0.2,nw_tos=0x00,icmp_type=8,icmp_code=0,actions=output:1
mininet>
```

11. We can change the default ovs-controller to another controller (Here we use pox as an example.)

11.1 open another terminal and kill the ovs-controller process

```
                    mininet@mininet-vm: ~
File  Edit  Tabs  Help
mininet@mininet-vm:~$ sudo ps -aux | grep ovs-controller
[sudo] password for mininet:
Sorry, try again.
[sudo] password for mininet:
root      3059  0.0  0.1  19372  1612 ?        S    10:19   0:00 ovs-controller ptcp:6633
mininet  4044  0.0  0.0   9388   932 pts/1    S+   10:22   0:00 grep --color=auto ovs-controller
mininet@mininet-vm:~$ sudo kill 3059
mininet@mininet-vm:~$
```

11.2 use dpctl to del-flows for switch s0

11.3 use dpctl to dump-flows for switch s0. To check that there are no rules for switch s0 now. So h1 and h2 can not ping h3.

```
mininet> s0 dpctl del-flows tcp:127.0.0.1:6634
mininet> s0 dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0x53fc8a0): flags=none type=1(flow)
mininet> h1 ping -c 1 h3
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

mininet>
mininet> h2 ping -c 1 h3
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 192.168.0.1 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

mininet>
```

12. Use pox controller.

```
mininet@mininet-vm:~$ cd pox/
mininet@mininet-vm:~/pox$ ./pox.py forwarding.l2_learning
POX 0.0.0 / Copyright 2011 James McCauley
DEBUG:core:POX 0.0.0 going up...
DEBUG:core:Running on CPython (2.7.3/Sep 26 2012 21:51:14)
INFO:core:POX 0.0.0 is up.
This program comes with ABSOLUTELY NO WARRANTY.  This program is free software,
and you are welcome to redistribute it under certain conditions.
Type 'help(pox.license)' for details.
DEBUG:openflow_of_01:Listening for connections on 0.0.0.0:6633
INFO:openflow_of_01:[Con 1/20941132740673] Connected to be-36-76-ec-c8-41
DEBUG:forwarding.l2_learning:Connection [Con 1/209411327406073]
Ready.
POX>
```

(above figure: the second terminal)

```
mininet> h1 ping -c 1 h3
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=63 time=34.2 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 34.286/34.286/34.286/0.000 ms
mininet> h2 ping -c 1 h3
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=63 time=44.8 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 44.802/44.802/44.802/0.000 ms
mininet>
```

(above figure: the first terminal)

```
mininet@mininet-vm:~/pox$ ./pox.py forwarding.l2_learning
POX 0.0.0 / Copyright 2011 James McCauley
DEBUG:core:POX 0.0.0 going up...
DEBUG:core:Running on CPython (2.7.3/Sep 26 2012 21:51:14)
INFO:core:POX 0.0.0 is up.
This program comes with ABSOLUTELY NO WARRANTY.  This program is free software,
and you are welcome to redistribute it under certain conditions.
Type 'help(pox.license)' for details.
DEBUG:openflow_of_01:Listening for connections on 0.0.0.0:6633
INFO:openflow_of_01:[Con 1/209411327406073] Connected to be-36-76-ec-c8-41
DEBUG:forwarding.l2_learning:Connection [Con 1/209411327406073]
Ready.
POX> DEBUG:forwarding.l2_learning:installing flow for 06:89:c0:40:58:73.1 -> da:4e:dc:57:63:9b.2
DEBUG:forwarding.l2_learning:installing flow for da:4e:dc:57:63:9b.2 -> 06:89:c0:40:58:73.1
DEBUG:forwarding.l2_learning:installing flow for 06:89:c0:40:58:73.1 -> da:4e:dc:57:63:9b.2
DEBUG:forwarding.l2_learning:installing flow for 06:89:c0:40:58:73.1 -> da:4e:dc:57:63:9b.2
DEBUG:forwarding.l2_learning:installing flow for da:4e:dc:57:63:9b.2 -> 06:89:c0:40:58:73.1
DEBUG:forwarding.l2_learning:installing flow for 06:89:c0:40:58:73.1 -> da:4e:dc:57:63:9b.2
POX>
```

(above figure: the second terminal)

Dr. Chih-Heng Ke

Department of Computer Science and Information Engineering, National Quemoy University, Kinmen, Taiwan

Email: smallko@gmail.com