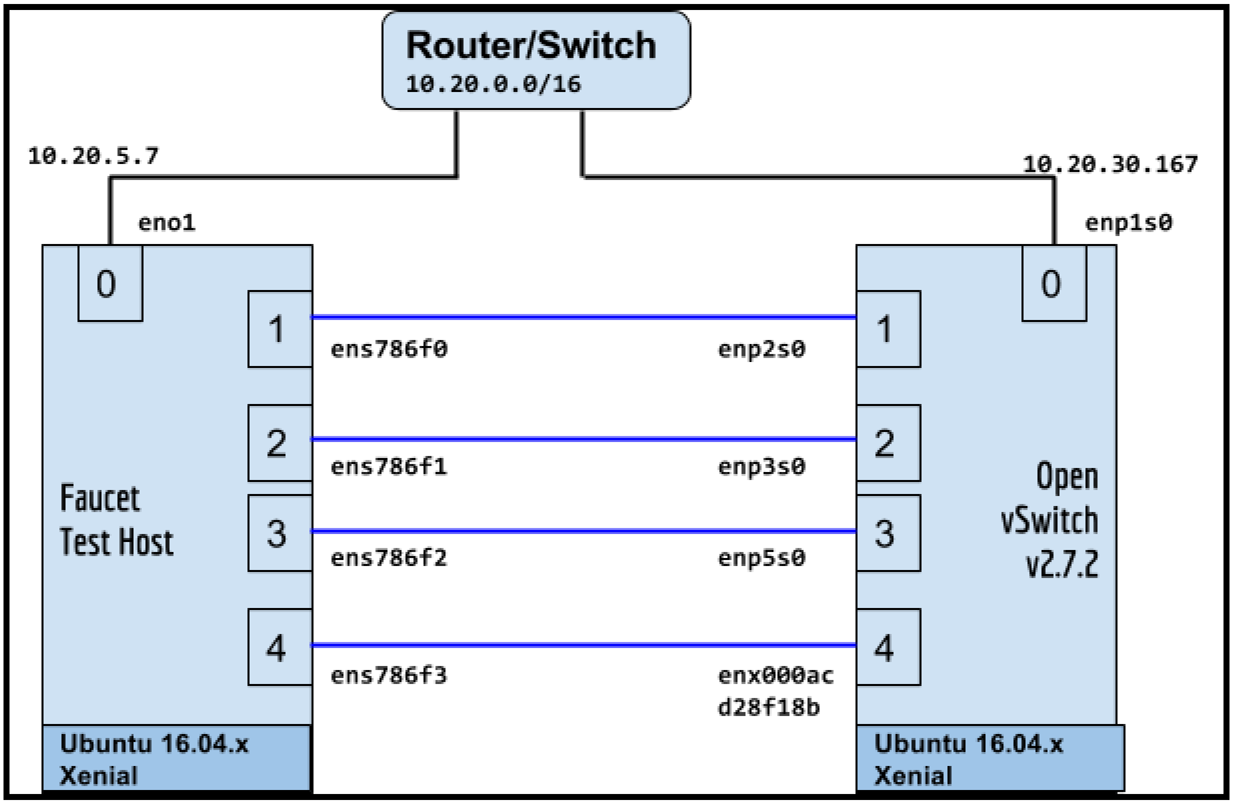
Faucet Testing with OVS on Hardware

Setup



Faucet Configuration file

```
# Faucet Configuration file: /etc/ryu/faucet/hw_switch_config.yaml
#
# If hw_switch value set to True, map a hardware OpenFlow switch to ports on this machine.
# Otherwise, run tests against OVS locally.
hw_switch: True
hardware: 'Open vSwitch'
# Map ports on the hardware switch, to physical ports on this machine.
# If using a switch with less than 4 dataplane ports available, run
# FaucetZodiac tests only. A 4th port must still be defined here and
# must exist, but will not be used.
dp_ports:
  1: ens786f0
  2: ens786f1
  3: ens786f2
  4: ens786f3

# Hardware switch's DPID
```

```
#dpid: 0xeccd6d9936ed
# ovs-sfcloud-qotom-1
dpid: 0xacd28f18b
#
# Port on this machine that connects to hardware switch's CPN port.
# Hardware switch must use IP address of this port as controller IP.
cpn_intf: eno1
# ovs-sfcloud-qotom-1
#cpn_intf: enx000acd28f18c
# There must be two controllers configured on the hardware switch,
# with same IP (see cpn_intf), but different ports - one for FAUCET,
# one for Gauge.
of_port: 6636
gauge_of_port: 6637
# If you wish to test OF over TLS to the hardware switch,
# set the following parameters per Ryu documentation.
# https://github.com/osrg/ryu/blob/master/doc/source/tls.rst
# ctl_privkey: ctl-privkey.pem
# ctl_cert: ctl-cert.pem
# ca_certs: /usr/local/var/lib/openvswitch/pki/switchca/cacert.pem
```

Hardware

1. For NICs, use Intel ones.
2. I have also used Hi-Speed USB to dual Ethernet which works great - http://vantecusa.com/products_detail.php?p_id=142&p_name=+USB+3.0+To+Dual+Gigabit+Ethernet+Network+Adapter&pc_id=21&pc_name=Network&pt_id=5&pt_name=Accessories
3. Once OVS is setup, use command

   ```
   # ovs-ofctl -O OpenFlow13 dump-ports-desc ovs-br0
   ```

   To make sure that Port speed is at least 1GB. If not, tests may not work correctly. (See Ethtool for more information)

Software

1. Ubuntu 16.04.x Xenial for OS
2. Open vSwitch 2.7.2 or 2.7.3 or 2.8.1

Commands

Faucet Test Host	**Open vSwitch**

Login as root on the Ubuntu system (16.04 used)
`# mkdir -p /usr/local/src/`
`# mkdir -p /etc/ryu/faucet/`
`# cd /usr/local/src/`
`# git clone https://github.com/faucetsdn/faucet.git`
`# cd faucet`
 	
`# ip a`
`1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc`
	Login as root on the Ubuntu system
	Install OVS v2.7.2 and start `openvswitch-switch` service
	
	`# systemctl status openvswitch-switch.service`
	
	`# ovs-vsctl add-br ovs-br0`
	
	`# ovs-vsctl add-port ovs-br0 enp2s0 -- set Interface enp2s0  ofport_request=1`

```
noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd
00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens786f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc mq state UP group default qlen 1000
    link/ether b4:96:91:00:88:a4 brd
ff:ff:ff:ff:ff:ff
    inet6 fe80::b696:91ff:fe00:88a4/64 scope link
       valid_lft forever preferred_lft forever
3: ens786f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc mq state UP group default qlen 1000
    link/ether b4:96:91:00:88:a5 brd
ff:ff:ff:ff:ff:ff
    inet6 fe80::b696:91ff:fe00:88a5/64 scope link
       valid_lft forever preferred_lft forever
4: ens786f2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc mq state UP group default qlen 1000
    link/ether b4:96:91:00:88:a6 brd
ff:ff:ff:ff:ff:ff
    inet6 fe80::b696:91ff:fe00:88a6/64 scope link
       valid_lft forever preferred_lft forever
5: ens786f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
1500 qdisc mq state UP group default qlen 1000
    link/ether b4:96:91:00:88:a7 brd
ff:ff:ff:ff:ff:ff
    inet6 fe80::b696:91ff:fe00:88a7/64 scope link
       valid_lft forever preferred_lft forever
6: ens802f0: <BROADCAST,MULTICAST> mtu 1500 qdisc
noop state DOWN group default qlen 1000
    link/ether 68:05:ca:3b:14:50 brd
ff:ff:ff:ff:ff:ff
7: ens787f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu
1500 qdisc mq state DOWN group default qlen 1000
    link/ether a0:36:9f:d5:64:18 brd
ff:ff:ff:ff:ff:ff
8: ens787f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu
1500 qdisc mq state DOWN group default qlen 1000
    link/ether a0:36:9f:d5:64:19 brd
ff:ff:ff:ff:ff:ff
9: ens787f2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu
1500 qdisc mq state DOWN group default qlen 1000
    link/ether a0:36:9f:d5:64:1a brd
ff:ff:ff:ff:ff:ff
10: ens787f3: <NO-CARRIER,BROADCAST,MULTICAST,UP>
mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether a0:36:9f:d5:64:1b brd
ff:ff:ff:ff:ff:ff
11: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
qdisc mq state UP group default qlen 1000
    link/ether 00:1e:67:ff:f6:80 brd
ff:ff:ff:ff:ff:ff
    inet 10.20.5.7/16 brd 10.20.255.255 scope global
eno1
       valid_lft forever preferred_lft forever
    inet6 cafe:babe::21e:67ff:feff:f680/64 scope
global mngtmpaddr dynamic
       valid_lft 86398sec preferred_lft 14398sec
    inet6 fe80::21e:67ff:feff:f680/64 scope link
       valid_lft forever preferred_lft forever
12: ens802f1: <BROADCAST,MULTICAST> mtu 1500 qdisc
noop state DOWN group default qlen 1000
    link/ether 68:05:ca:3b:14:51 brd
ff:ff:ff:ff:ff:ff
```

```
# ovs-vsctl add-port ovs-br0 enp3s0 -- set Interface
enp3s0  ofport_request=2
# ovs-vsctl add-port ovs-br0 enp5s0 -- set Interface
enp5s0  ofport_request=3
# ovs-vsctl add-port ovs-br0 enx000acd28f18b -- set
Interface enx000acd28f18b  ofport_request=4

# ovs-vsctl set-fail-mode ovs-br0 secure
# ovs-vsctl set bridge ovs-br0 protocols=OpenFlow13
# ovs-vsctl set-controller ovs-br0
tcp:10.20.5.7:6636 tcp:10.20.5.7:6637

# ovs-vsctl get bridge ovs-br0 datapath_id

# ovs-vsctl show
308038ec-495d-412d-9b13-fe95bda4e176
    Bridge "ovs-br0"
        Controller "tcp:10.20.5.7:6636"
        Controller "tcp:10.20.5.7:6637"
        Port "enp3s0"
            Interface "enp3s0"
           Port "enp2s0"
            Interface "enp2s0"
         Port "enx000acd28f18b"
            Interface "enx000acd28f18b"
        Port "ovs-br0"
            Interface "ovs-br0"
                type: internal
        Port "enp5s0"
            Interface "enp5s0"
                type: system
    ovs_version: "2.7.0"

# ovs-vsctl -- --columns=name,ofport list Interface
name                : "ovs-br0"
ofport              : 65534

name                : "enp5s0"
ofport              : 3

name                : "enp2s0"
ofport              : 1

name                : "enx000acd28f18b"
ofport              : 4

name                : "enp3s0"
ofport              : 2
```

To locate the corresponding physical port, you can make the port LED blink. For example:

```
# ethtool -p enp2s0 5

# ovs-ofctl -O OpenFlow13 dump-ports-desc ovs-br0
OFPST_PORT_DESC reply (OF1.3) (xid=0x2):
 1(enp2s0): addr:00:0e:c4:ce:77:25
     config:     0
     state:      0
     current:    1GB-FD COPPER AUTO_NEG
     advertised: 10MB-HD 10MB-FD 100MB-HD 100MB-FD
1GB-FD COPPER AUTO_NEG AUTO_PAUSE
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD
1GB-FD COPPER AUTO_NEG AUTO_PAUSE
     speed: 1000 Mbps now, 1000 Mbps max
 2(enp3s0): addr:00:0e:c4:ce:77:26
     config:     0
```

```
13: eno2:
<NO-CARRIER,BROADCAST,MULTICAST,PROMISC,UP> mtu 1500
qdisc mq state DOWN group default qlen 1000
    link/ether 00:1e:67:ff:f6:81 brd
ff:ff:ff:ff:ff:ff
    inet6 cafe:babe::21e:67ff:feff:f681/64 scope
global mngtmpaddr dynamic
       valid_lft 82943sec preferred_lft 10943sec
    inet6 fe80::21e:67ff:feff:f681/64 scope link
       valid_lft forever preferred_lft forever
16: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu
1500 qdisc noqueue state DOWN group default
    link/ether 02:42:40:9d:0d:65 brd
ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:40ff:fe9d:d65/64 scope link
       valid_lft forever preferred_lft forever
```

To locate the corresponding physical port, you can make the port LED blink. For example:

```
# ethtool -p ens786f0 5

# cp
/usr/local/src/faucet/tests/hw_switch_config.yaml
/etc/ryu/faucet/hw_switch_config.yaml
```

Edit the hw_switch_config.yaml file as shown earlier in this document. But, set the hw_switch=False

```
# cd /usr/local/src/faucet/
# apt install docker.io
# docker build -t faucet/tests -f Dockerfile.tests .
# apparmor_parser -R
/etc/apparmor.d/usr.sbin.tcpdump
# modprobe openvswitch
# docker run --privileged --net=host -v
/etc/ryu/faucet:/etc/ryu/faucet -v /tmp:/tmp -ti
faucet/tests
```

Once the above minitest version is successful, then edit the `hw_switch_config.yaml` file as shown earlier in this document. But, set the `hw_switch=`**`True`**

```
# docker run --privileged --net=host -v
/etc/ryu/faucet:/etc/ryu/faucet -v /tmp:/tmp -ti
faucet/tests
```

Test Results:
100% of tests MUST pass. For up-to-date information on test runs, check out Travis Status page @ - https://travis-ci.org/faucetsdn/faucet

```
     state:      0
     current:    1GB-FD COPPER AUTO_NEG
     advertised: 10MB-HD 10MB-FD 100MB-HD 100MB-FD
1GB-FD COPPER AUTO_NEG AUTO_PAUSE
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD
1GB-FD COPPER AUTO_NEG AUTO_PAUSE
     speed: 1000 Mbps now, 1000 Mbps max
 3(enp5s0): addr:00:0e:c4:ce:77:27
     config:     0
     state:      0
     current:    1GB-FD COPPER AUTO_NEG
     advertised: 10MB-HD 10MB-FD 100MB-HD 100MB-FD
1GB-FD COPPER AUTO_NEG AUTO_PAUSE
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD
1GB-FD COPPER AUTO_NEG AUTO_PAUSE
     speed: 1000 Mbps now, 1000 Mbps max
 4(enx000acd28f18b): addr:00:0a:cd:28:f1:8b
     config:     0
     state:      0
     current:    1GB-FD COPPER AUTO_NEG
     advertised: 10MB-HD COPPER AUTO_NEG AUTO_PAUSE
AUTO_PAUSE_ASYM
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD
1GB-HD 1GB-FD COPPER AUTO_NEG
     speed: 1000 Mbps now, 1000 Mbps max
 LOCAL(ovs-br0): addr:00:0a:cd:28:f1:8b
     config:     PORT_DOWN
     state:      LINK_DOWN
     speed: 0 Mbps now, 0 Mbps max
```

Debugging

TCPDump

Many times, we want to know what is coming in on a port. To check on interface "`enp2s0`", for example, use

```
# tcpdump -A -w enp2s0_all.pcap -i enp2s0
```

Or

```
# tcpdump -A -w enp2s0_all.pcap -i enp2s0 ‘dst host <controller-ip-address> and port 6653’
```

To read the pcap file, use

```
# tcpdump -r enp2s0_all.pcap
```

More detailed examples are @ https://www.wains.be/pub/networking/tcpdump_advanced_filters.txt

Note:

Q: On which machine should one run **`tcpdump`**?

A: Depends. If you want to understand for example, what packet_ins are sent from switch to controller, run on switch side on the interface that is talking to the controller. If you are interested on what is coming on a particular test port, then run it on the Test Host on that interface.

Ethtool

- To locate a physical port say enp2s0, make the LED blink for 5 seconds:

  ```
  # ethtool -p enp2s0 5
  ```

- To figure out speed on the interface. Note that if Speed on the interface is at least not 1G, then tests may not run correctly.

  ```
  # ethtool enp2s0
  # ethtool enp2s0 | grep Speed
  ```

  Reference: https://www.garron.me/en/linux/ubuntu-network-speed-duplex-lan.html