Research and Education Advanced
Network New Zealand

A Manual for OpenFlow Test Tool for Faucet/Valve

Author:
Se-young Yu

Supervisor:
Chris Lorier

May 12, 2016

1 Introduction

This Maual explains what is OpenFlow Test Tool for Faucet/Valve and how to use it. This work is an outcome of REANNZ Future Funds summer internship project in March - May 2016.

2 Motivation

There have been many OpenFlow enabled switches (i.e. bare metal switch) made available from different manufacturers, however their OpenFlow implementation on the switch differs for each manufacturer.

To ensure an OpenFlow enabled switch to run any OpenFlow controller application, the user needs to know the following.

1. What OpenFlow version does the switch supports and controller is running on?
2. Which OpenFlow instructions does the switch supports?
3. Does the consecutive OpenFlow instructions imposed by controller works on the switch?

There are tools to do 1 and 2, but it is not easy to find a tool to test whether a series of OpenFlow instructions used in an OpenFlow controller application works on a switch, because each OpenFlow application has different work flow.

3 Testbed

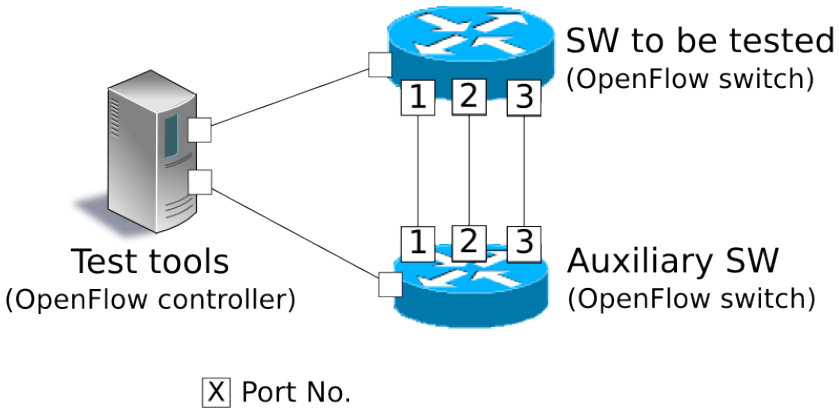

Figure 1: Testbed setup[1]

Figure 1 shows the setup of the testbed. The test tool runs on a testbed consist of two OpenFlow switches, an auxiliary switch and a target switch, and one OpenFlow controller application. The tester application, which is an RYU application runs on the controller application, sends OpenFlow commands to the two switches to initiate a test and validate the result.

Two OpenFlow switches needs to be connected to the OpenFlow controller running on the tester machine. Both switches can be either physical switch or OpenVSwitch. To test a physical switch, it is easier to have OpenVSwitch and tester application running on the same machine. A script[2] to set up this testbed is included in the tool.

4 Test Procedure

This section describes test procedure of the OpenFlow Test Tool. For the actual implementation of the test tool, please refer to `https://github.com/`

[1]The image is imported from `https://osrg.github.io/ryu-book/en/html/switch_test_tool.html`

[2]`https://github.com/youf3/faucet/blob/sy-testtool/tests/testbed-setting.sh`

youf3/faucet/tree/sy-testtool/tests

4.1 Test Execution

To execute a test, the testbed needs to be set up as described in Section 3. Once both switches and controller is configured, the following command is used to run the test tool.

"ryu-manager –test-switch-dir ./packets –test-switch-target [target-dpid] –test-switch-tester [tester-dpid] ./tester.py"

[target-dpid] and [tester-dpid] are dpid for auxiliary and target switch to be tested.

4.2 Test pattern

For each test, a test pattern is required to specify details of the test. A test pattern is in a JSON format, which consists of a "description", "prerequisite" and "tests". "description" is a string describing what this test is about.

"prerequisite" specifies OpenFlow Flow State Modification message that needs to be installed before sending actual packets, in the form of RYU's FlowMod object representation[3]. For each "prerequisite", it is required to have "OXMTlv" object with "in_port" field and its value to specify the port which the packet will be received from the auxiliary switch. In addition to the "in_port", it also need "OFPActionOutput" object specified in "OFPInstructionActions" to specify which port the packet needs to be forwarded in the target switch.

"tests" specifies actual packet data and flow control of packets to be generated from auxiliary switch. Each "tests" is consist of "ingress" and "egress". "ingress" specifies the packet data using RYU's packet object using JSON representation[4]. "egress" is used to check against the packet forwarded from the target switch. It specifies packet data expected to receive from the target switch.

[3] http://ryu.readthedocs.io/en/latest/ofproto_v1_3_ref.html?highlight=flowmod#ryu.ofproto.ofproto_v1_3_parser.OFPFlowMod

[4] https://osrg.github.io/ryu-book/en/html/packet_lib.html#basic-usage

4.3 Testing Single Packet-in and Packet-out

The single packet-in and packet-out test is to test the target switch with a user-specified packet sent to the specific port. This test ensures the switch and Faucet are able to recognise packets user specified and forward the packets to the right port.

The user can specify packet header and packet payload to be generated using "ingress" and "egress" in the test pattern. To check whether the packet forwarded from the target switch is identical to the packet sent to the switch, keep the "ingress" fields and "egress" fields identical. Refer to `https://github.com/youf3/faucet/blob/sy-testtool/tests/switch_test/packets/packet_in_tcp.json` for an example on how to compose a test pattern with user-specified packet-in and packet-out test.

4.4 Testing a stream of packets

In addition to the single packet test, the tool can test switch with a continuous stream of user-specified packet. The user can specify throughput of packets from 1 Kb/s to an arbitrary number, however generating throughput more than 100 Mb/s is not efficient due to the design of the packet_out action message implementation in RYU.

Variables "pktps" and "duration_time" are used to specify throughput and the duration of the packet stream in terms of packet/s and seconds. The default value for "pktps" is 1000 packet/s and "duration_time" is 30 seconds. These keyword should be instance variables in a "packets" object in a test pattern. When they are used, "egress" object in "test" should contain "throughput" object which contains "OFPMatch" object and "kbps" variable. The "kbps" variable is used to check throughput of packets forwarded from the target switch. If the target switch is expected to forward all packets without dropping any, then the value of "kbps" should be computed as $pktps \times packet\ size$. For more detail and examples, refer to `https://github.com/youf3/faucet/blob/sy-testtool/tests/switch_test/packets/50M_IPv6_TCP.json`

4.4.1 Configuration Reloading Test

When testing a stream of packets with a target switch, "reload" variable can be set to test what happens when you reload a configuration in Faucet at

the middle of the test. The "reload" variable is an optional boolean variable in "packets" object. When it is set to true, tester will reload Faucet configuration at the middle of the test to check how many packets lost during the reloading. For more detail and examples, refer to `https://github.com/youf3/faucet/blob/sy-testtool/tests/switch_test/packets/reload_conf.json`

5 Reporting a Test Result

The test tool produce a simple report after tests. It specifies whether each test has passed or failed, based on the "egress" field in the test pattern. For a Single packet-in and packet-out test, the contents and out_port of the packet forwarded will be check upon, and for packet stream test throughput and out_port of the packet stream will be checked.

In addition to the pass/fail status, it will also report number of packet lost during the reloading test, if the reloading configuration is tested.

6 Future works

Because the tool is developed to test basic functionality of Faucet/Valve, the test tool calls Faucet/Valve functions to evaluate its outcome. This makes the tool tightly coupled with Faucet/Valve with OpenFlow 1.3 and may not be used with other OpenFlow controller applications/versions. To use this tool to test other OpenFlow controller application, subsequent part of the test tool needs to be re-written as well as the controller application itself.

The test tool is capable of testing Faucet/Valve with a stream of packets, however the maximum throughput of the packet generation is inefficient. In our testbed, we only achieved a maximum of 100 Mb/s of packet until the test tool start dropping significant amount of packets.