# The State of OpenFlow 2012
# Report and Analysis

In partnership with

Report RA-011513-01
January 2013

# The State of OpenFlow 2012 Report and Analysis

PRODUCTS TESTED:
PICA8 PRONTO 3290
HP PROCURVE 6200YL
IWNETWORKS SDN 8950

*This Report would not be possible without the support of the following people:*

Christian Esteve Rothenberg, Ph.D. for his support in utilizing RouteFlow as one of our testing tools.

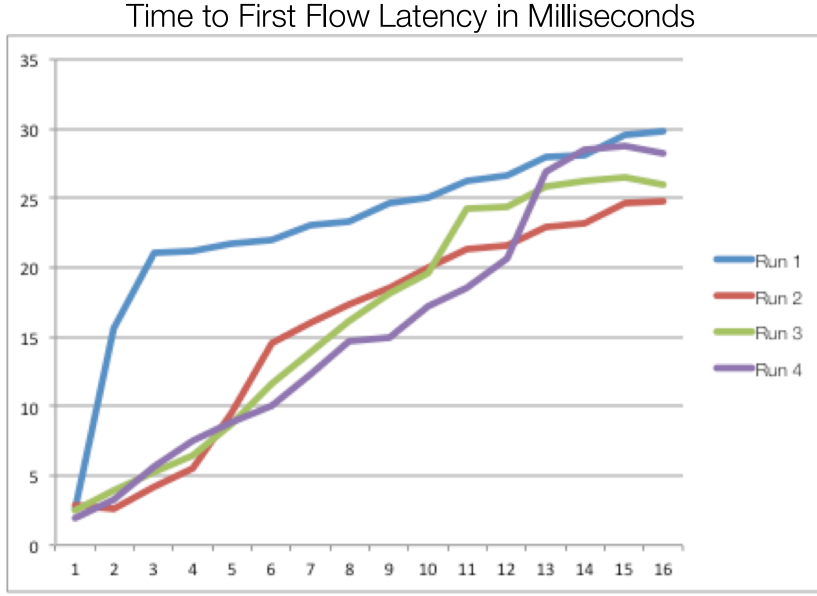
John Furrier of SiliconANGLE and Stuart Miniman of Wikibon for their support of open data and Router Analysis, Inc.

And finally the Open Source community without whom we would not have anything to test.

Summary:

- OpenFlow v1.x shows promise and has the support to become part of the network device decision tree in 2013.
- Vendors like Pica8 and HP are pushing hard to show the value of OpenFlow programmable switches in the Enterprise network.
- Depending on the application, latency for the Time to First Flow can be high and may need to be addressed.

Time to First Flow Latency in Milliseconds




TCP Packets as received in order of lowest latency to highest latency

# Table of Contents



# Table of Figures

## Introduction by John Furrier of SiliconANGLE

Welcome to the State of OpenFlow 2012 brought to you by SiliconANGLE in partnership with Router Analysis, Inc. Software-defined Networking (SDN) is far from defined. It is just the beginning and it is being led by software or what we call Software-led Infrastructure (SLI). SLI and SDN are exploding with innovation from the hottest startups to the biggest networking vendors.

The market is scrambling to understand and get a position in this highly important emerging market. OpenFlow innovation has been moving very quickly and is positioned to succeed in 2013.

SiliconANGLE, Wikibon and Router Analysis have been committed to analysis of key emerging areas and having a testing lab to "kick the tires" with the leading and emerging new companies will allow IT executives and engineers make the most informed decisions in this important area of innovation of Software-led Infrastructure (SLI).

## Statement from Steven Noble, founder of Router Analysis, Inc.

The tests results contained within this document are to the best of my knowledge accurate representations of the products tested. In order to keep vendor neutrality, no vendors were allowed in the Router Analysis Testing Lab during the tests. The testing to produce this report was done for free by Router Analysis, Inc. Other than the switch provided by IWNetworks, all equipment used for testing is owned by Router Analysis, Inc. or provided by neutral third parties.

The State of OpenFlow 2012 report is being released to both vendors and end users at the same time; no one other than John Furrier, Christian Rothenberg or myself knows exactly what is contained in the report.

Any questions can be addressed directly to Router Analysis, Inc. via the email address info@routeranalysis.com.

## Testing Session Overview

The goal of the testing was straightforward; confirm that currently available switches interoperate with two of the most used OpenFlow controllers, Floodlight and NOX (using the RouteFlow project).

The minimum requirements for the switches were set low on purpose:

- Minimum of 1000 Flows in Hardware
- Support of OpenFlow v1.0
- Interact with both Floodlight and RouteFlow

To keep the tests fair support for higher versions of OpenFlow was ignored, as were HP's extensions such as priority queue support.

All three vendors who participated in our State of OpenFlow 2012 test passed the basic features tests.

## Vendor Involvement

Router Analysis, Inc. indirectly via Twitter, Blog posts and word of mouth invited any vendor with an OpenFlow v1.0 capable switch to be involved in the testing.

One vendor, IWNetworks contacted Router Analysis, Inc. to take up the public offer.

We directly invited the following vendors:

- Juniper – Unable to participate – Still in limited testing
- Cisco – Unable to participate – Still in limited testing
- Pica8 – Participated and provided support
- HP – Responded by did not provide an evaluation unit – Router Analysis, Inc. purchased a 6200yl for the tests
- Arista – No Answer
- Brocade – Responded and Declined

Total number of vendors invited: 6, total participating: 3.

### Analysis by John Furrier of SiliconANGLE

Router Analysis contacted six (6) tier-1 companies selling networking products including emerging startups selling new SDN products. One thing that was interesting was who responded and who did not. We cannot comment on or analyze what was not available. It is our assumption that no response means that the vendor did not have a product that fit our test criteria.

## Test Results Summary

Pica8's claim of up to 4k flows in hardware on the Pronto 3290 was confirmed. Both HP and IWNetworks were able to handle above the minimum 1000 flows but neither claimed a number. All three vendors were able to handle line-rate traffic for the flows once installed.

When testing ease of setup, Hybrid mode setup without any documentation or help was accomplished with all three vendors. In OpenFlow only mode, Pica8's directions were used to setup their switch. While still appearing the same, HP offers a command to turn all ports into OpenFlow Ports. The IWNetworks switch did not have a non-hybrid code available.

The most complex part of the testing was troubleshooting flow insertion. For example: In Hybrid mode, Pica8 required the VLAN to be set and would reject any flows that did not match the OpenFlow enabled VLAN associated. HP required the controller connection to be in a separate VLAN and tweaks to the memory space.

Once each vendor's specific needs were worked around, scale testing proceeded.

Latency across all switches was acceptable at about 3 microseconds once flows were installed. HP's number was higher, but due to the age of the forwarding chipset in the 6200yl switch tested, this would be expected.

TTFF (Time to First Flow) varied and requires more investigation. Our observation is that times between 20 and 30 milliseconds appear to be the most common. While doing a TTFF study with Christian Esteve Rothenberg, Ph.D., TTFF times as low as 1 millisecond and as high as 47 milliseconds were seen. These RTT times to the external control plane depend on the controller implementation of choice and are application specific.

When setting up multiple flows at the same time, it appears that somewhere in the path between the switch and the controller, packets were being queued and processed serially. As there are many moving parts in the destination controller application design (see Figure 2) and the test was only attempted using the Pica8 Pronto 3290, further testing and study is necessary to pin down the issue.

## Testing Flow Installation Times

Flow installation times for 1000 flows were about five seconds for both Pica8 and IWNetworks, i.e. about 200 flow insertions/second. There was a slight variations on the time to get maximum flows into the HP 6200yl based on how flow table space was allocated.

The numbers for the HP 6200yl with the minimum amount of table space allocated to OpenFlow to fit 1000 flows took about seven seconds. When all of the table space was allocated to OpenFlow flow installation time came in around nine seconds. Future testing will include higher numbers of flows, OpenFlow agent contention and different ways of feeding the flows into the devices to try and get more specific numbers.

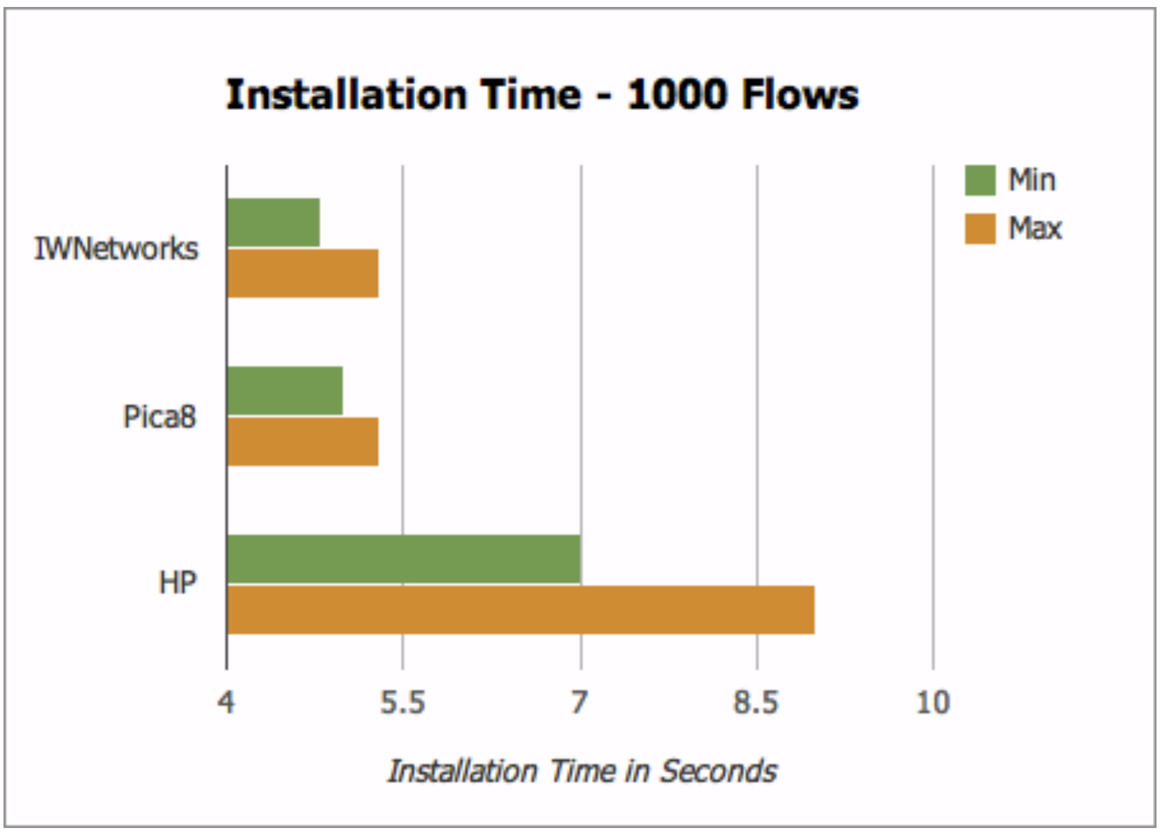



Figure 1 – Installation time for 1000 flows.

## Test Lab Setup

The *Router Analysis* test lab utilizes specially built systems from IXSystems. The hardware specs are the following:

SuperMicro X9SAE-V with a single Intel I7-3770 CPU, 32GB ECC RAM, 240GB Intel 520 Series SSD and 4 Intel I340-T2 GbE NICs.

The Server was configured with VMware 5.1.0 and both Floodlight and RouteFlow were given VMs with 4G RAM and 2vCPUs. For RouteFlow we assigned 3 GbE interfaces to the testing. One Interface was connected between the "base" system where RFServer and NOX run and one interface was assigned to each end host (end hosts were run in LXC containers on the same VM).

For latency measurements and tests utilizing more than two hosts Router Analysis uses our own proprietary testing system.

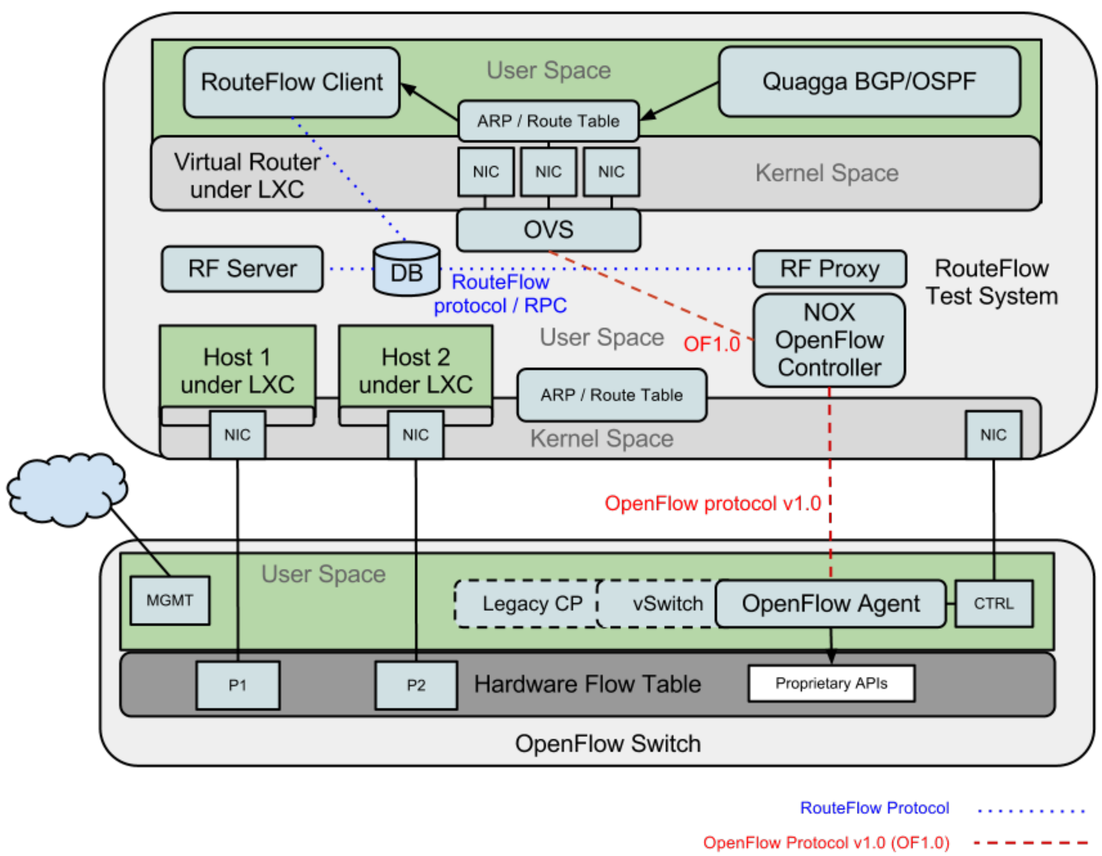



Figure 2 – Diagram of Test Setup with Two Ports

## Tests Carried Out

- OpenFlow setup and configuration
    - Ease of Configuration
        - Without Documentation
        - With Documentation
        - With Research and/or Vendor Support
    - Post Configuration Details
        - Status of Connection to Controller
        - Status of Flows Installed
- Interoperability with BigSwitch Floodlight
    - Floodlight Alone
    - Floodlight with 1000 Flows from RA Flow Tool
- Interoperability with NOX
    - NOX Alone
- Interoperability with RouteFlow
    - With Two Hosts
    - With Eight Hosts
    - Latency Data

## Testing Results In-Depth

### Configuring OpenFlow on the switches in Hybrid Mode

Configuring each of the switches was straightforward in Hybrid mode. For example to configure the Pica8, the following configuration was put into the CLI:

```
open-flow {
    fail-mode: "standalone"
    controller floodlight {
        protocol: "tcp"
        address: 192.168.10.114
        port: 6633
    }
```

The configuration says "Use 192.168.10.114 as the controller, speak TCP to port 6633. In the event that you lose connectivity to the controller, go into standalone switch mode"

The same configuration on the HP looks like this:

```
openflow
   enable
   controller-id 1 ip 192.168.10.114 controller-
interface vlan 10
   instance "floodlight"
      listen-port
      member vlan 1
      controller-id 1
      connection-interruption-mode fail-standalone
      enable
      exit
   exit
```

In HPs case, VLANs need to be assigned to use within the OpenFlow configuration, other than that, the configuration does the same as the one on the Pica8 switch.

The IWNetworks switch is even more straightforward:

```
openflow controller 192.168.10.114 6633 tcp
openflow default-table full-match
openflow enable
```

The configuration of all three vendors in Hybrid mode was done using the CLI help system i.e. “?”. Each switch used openflow as the root configuration keyword (PicaOS uses open-flow) and the commands were determined after that.

### Configuration in OpenFlow Only Mode

On the HP, the same CLI was used to configure the switch so that all ports as ran as OpenFlow. No specific differences were seen between the two modes.

The IWNetworks code supplied only supported Hybrid mode.

For the Pica8, we booted the system into a shell from the console boot prompt and had to run commands by hand (we eventually used a script provided by Josh Bailey of Google to automate the configuration).

The Pica8 configuration in OpenFlow only mode is much like configuring OVS on a standard Linux PC. Pica8 has done a great job with their reference documents available at http://pica8.com/blogs/?p=329.

### Floodlight Interoperability

Floodlight was run on an Ubuntu Linux VM with the 2.6.38-16-generic kernel. Floodlight was started with the following command “java –jar floodlight-0.90.jar”.

The switches were configured on a separate network from the Floodlight controller and made their connection over tcp to port 6633 of the controller.

Once the switches were connected and seen by floodlight, the Router Analysis Flow Tool was used to insert 1000 flows into the switches. The flows contained source and destination addresses and ports.

When the test tool was run the following information was seen:

```
New flow from 149.126.72.248:58164 to 193.72.216.2:53
{"status" : "Entry pushed"}
```

The flow was confirmed on the switch:

```
cookie=0xffffffffec6a7509, duration=54.033s, table=0,
n_packets=0, n_bytes=0,
priority=32767,ip,dl_vlan=100,nw_src=149.126.72.248,nw_dst=
193.72.216.2 actions=output:5
```

## NOX Interoperability

As NOX is also part of RouteFlow it can be assumed that there would be no issue with interoperability. With NOX running stand-alone, the proper base "to controller" flows were inserted into the systems.

## RouteFlow Interoperability

The RouteFlow project by CPqD builds a full OpenFlow based routing stack. RouteFlow utilizes Quagga running in a virtual machine, along with either NOX or POX as the OpenFlow controller (Floodlight and RYU are also supported in other releases) and the RFServer/RFProxy programs allowing data to be passed between the Quagga router and the OpenFlow controller.

RouteFlow testing involved three different scenarios.

### Scenario 1:

The first scenario used the default setup running on Ubuntu with Quagga running in a LXC container named rfvm1 and two host machines running in their own LXC containers named b1 and b2.

The Network Interfaces were setup in the following way:
eth0 was connected to the switch as the controller connection.
eth1 was used as a management interface on the local lan
eth2 was used for connectivity to b1, eth3 for b2.
Here is how the HP 6200yl was configured:

```
openflow
   enable
controller-id 2 ip 192.168.1.17 controller-interface vlan 2
   instance "routeflow"
      listen-port
      member vlan 1
      controller-id 2
      connection-interruption-mode fail-standalone
      enable
      exit
exit
```

Above, a new controller, controller 2, is configured. A new OpenFlow instance named routeflow is created and configured.

We had to add a new VLAN, VLAN 2

```
vlan 2
   name "CNTL_VLAN"
   untagged 2
   ip address 192.168.1.100 255.255.255.0
   exit
```

This VLAN is used to communicate with the controller and based on HP's configuration guide must be separate from the OpenFlow enabled VLAN.

### Scenario 2:

In the second scenario, six more hosts were added for a total of eight. The six new hosts, running on the Router Analysis Testing Platform had the ability to generate line-rate traffic and were used to confirm that once flows were installed, all boxes could pass traffic.

There were no issues in the second scenario. All hosts were visible and able to pass traffic between each other.

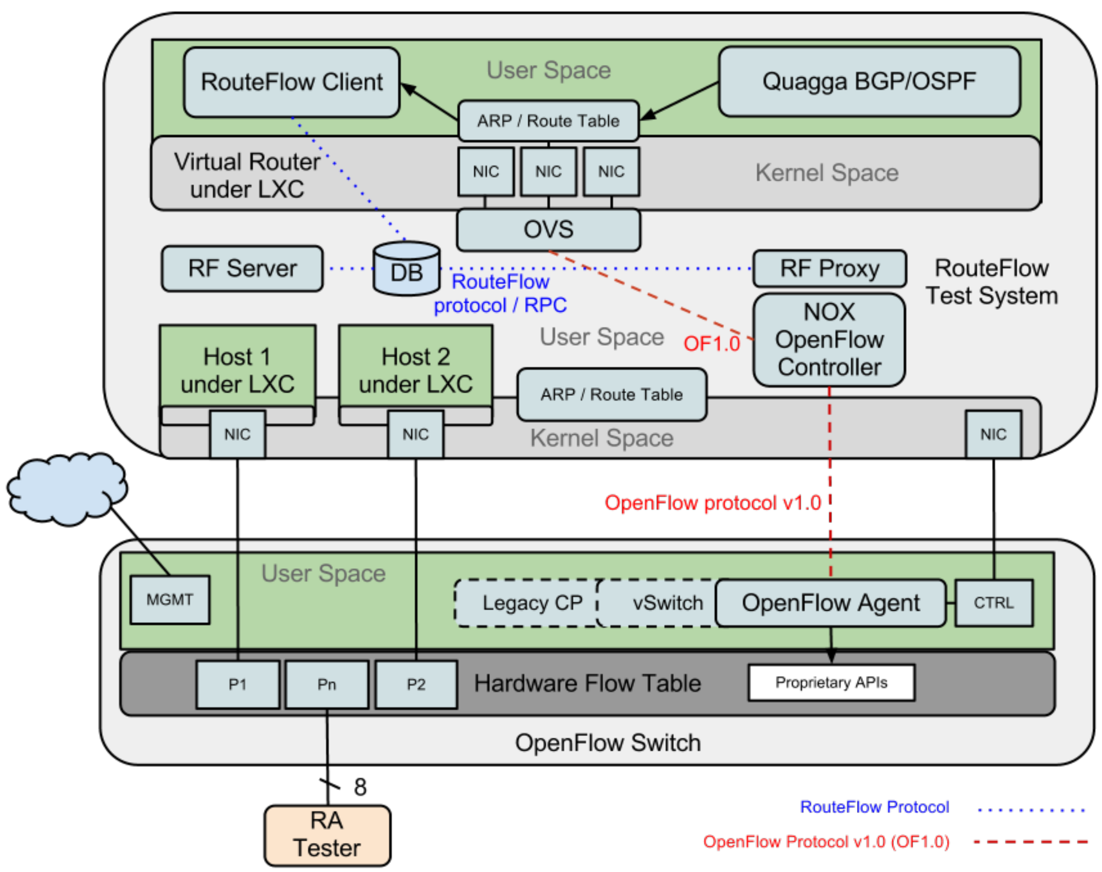



Figure 3 – RouteFlow Testing Scenario 2 and 3

### Scenario 3:

The third scenario used Router Analysis' own test tools to generate and measure the latency of Time to First Flow. Unidirectional TCP packets were used for the measurements in this test.

When doing the testing TCP packets were tagged with a signature and time stamp. Then a single TCP packet was generated and sent to another host where the time stamp was grabbed and the TTFF was determined. To confirm that packets were being sent up to the controller a tap was on the link between the switch and the controller to monitor for our signed packet.

The results came out a bit different than expected as all previous testing using PING latency timing was done using a single host at a time. In the TCP based testing we were able to simulate multiple flows coming in at exactly the same time.

TTFF times in the 1 millisecond range were seen on occasion, along with numbers in the 40 millisecond range. As these packets were sent in one direction, the results should be different than the ICMP numbers, but still within the same spacing.

When testing sending multiple packets at the same time, the TTFF of each packet increase serially. In the short time spent debugging the issue it was clear that either the switch or controller was sending or processing packets in a serial rather than parallel. This caused latency times to go up as more packets were sent in parallel.

During testing, sets of 1, 2, 3, 4 and 8 packets were sent at the same time on different links and the timing was recorded. In the final tests16 packets were sent per test, 2 per link from different source and destination addresses.

The graph in figure 4 shows ten TCP test runs of eight packets each. To simplify the graph, the runs were limited to tests which had initial TTFF numbers between 1 and 22 milliseconds. From the graph we can see that TTFF appears to level out as you get more requests.

Router Analysis will be doing more testing in this space in the future.

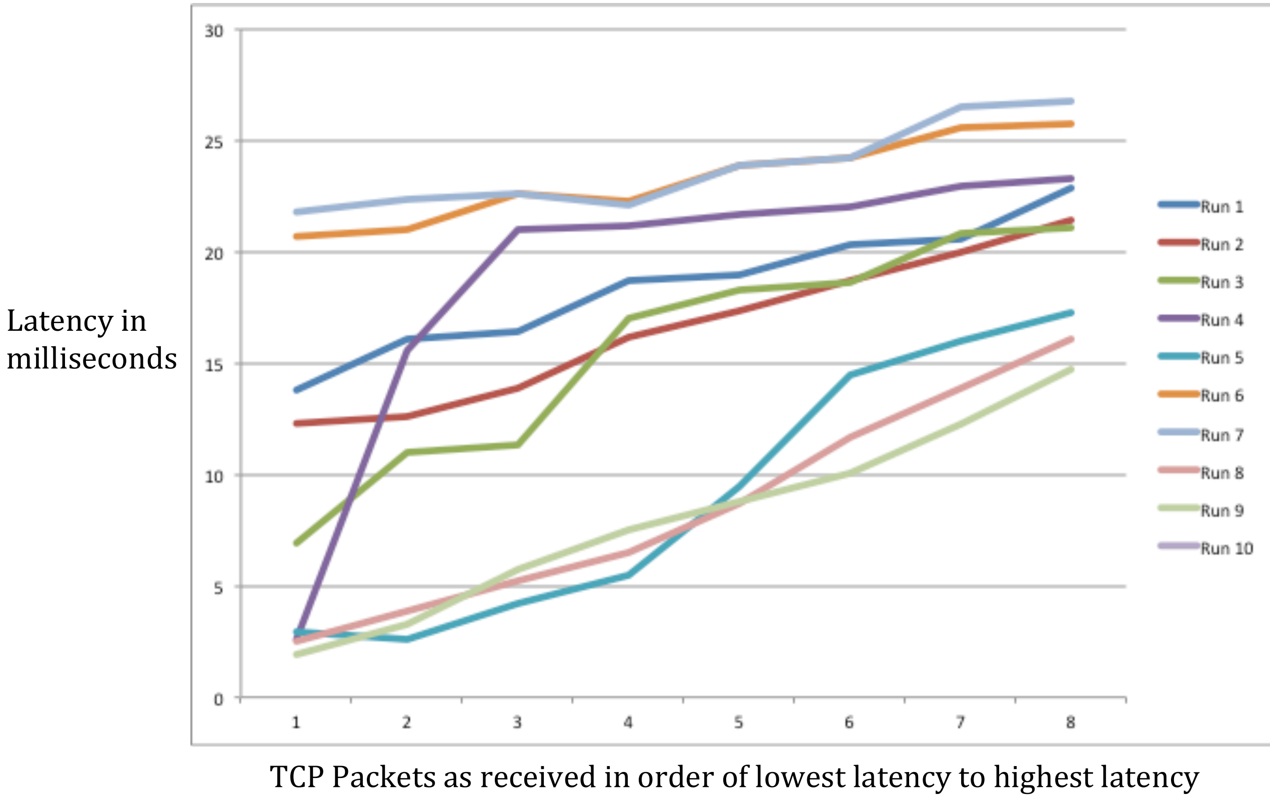



**Figure 4 - TTFF Latency**

Note:

In order to best understand what the expected results should look like, ICMP packets were first used to measure the round trip time between hosts 1 and 2. These numbers are different than numbers that would be seen when sending ICMP packets from the Virtual Router on the reference host (see figure 2 or 3) as packets from the Virtual Router will always pass through the controller.

For ICMP RTT when no flow existed, TTFF metrics of between 10 and 30 milliseconds were observed for the first packet and then a consistent 700 microseconds after that.

## Final Thoughts from Steven Noble

With companies like Pica8 putting a large amount of resources behind OpenFlow enabled switches and HP's statements about having millions of OpenFlow enabled ports deployed we see OpenFlow as a serious contender in 2013.

That being said, during the two weeks Router Analysis spent testing OpenFlow on multiple switches, issues were found that will require more in-depth testing and study. Time to First Flow was an eye-opening test.

Measuring TTFF using both unidirectional TCP streams and RTT ICMP packets gave the same basic information: The number is not stable and depends on many variables. Router Analysis plans to do isolation testing in the future using to better determine where the jitter and latency comes from.

When sending many flows at the same time there appears to be serial processing by the controller and/or OpenFlow agent implementation in the switch CPU. Some results show the first few flows all taking the same amount of time. Other results had numbers that increased linearly 2-4 milliseconds at a time.

Past the TTFF issue, Router Analysis saw functional and stable products from all vendors who participated.

The support from Pica8 was outstanding and led to us using the Pronto 3290 as our reference switch. The Pronto 3290 also had the largest flow memory of any of the switches tested.

For latency numbers, IWNetworks was lowest. IWNetworks provided Router Analysis, Inc. with their newest product and the only switch in the test that supported 40GbE.

As HP did not supply a switch, Router Analysis had to go out and purchase one. The ProCurve 6200yl was what Router Analysis was able to acquire that supported OpenFlow. In discussions with HP employees, Router Analysis was informed that the 6200yl is not the best example of OpenFlow enabled HP switches, but no other products were offered.

Orchestration is going to be key to the migration from the hardware datacenter to the Software-led one. With projects like Network Function Virtualization being broken out of SDN and taking on a life of their own, we see Network API controlled hardware as being vital. OpenFlow provides enough functionality to be used as a migration tool and later as part of an orchestration controller.

# Appendix A – Switch Configurations

These are the switch configurations used in the *Router Analysis* lab.

## HP 6200yl

### Version:

HP-6200yl-24G# show ver
Aug 24 2012 09:30:03
K.15.10.0003

### OpenFlow:

```
openflow
   limit policy-engine-usage 100
   enable
   controller-id 1 ip 192.168.10.114 controller-interface
vlan 10
   controller-id 2 ip 192.168.1.17 controller-interface
vlan 3
   instance "routeflow"
      listen-port
      member vlan 1
      controller-id 2
      connection-interruption-mode fail-standalone
      enable
      exit
   instance "floodlight"
      connection-interruption-mode fail-standalone
      exit
   exit
```

### VLAN Configuration:

```
vlan 1
  name "DEFAULT_VLAN"
  no untagged 1
  untagged 2-24
  ip address dhcp-bootp
  exit
vlan 10
  name "MGMT_VLAN"
  untagged 1
  ip address 192.168.21.101 255.255.255.0
  exit
```

# Appendix B – Test Result Data Tables

The following tables represent a sample of the data collected during testing in the *Router Analysis* Lab. If you would like more data or information on how results were determined please contact Router Analysis.

## Number of flows each switch handles by default

| Switch | Hybrid Mode | OpenFlow Only Mode |
|---|---|---|
| HP 6200yl | 1373 | 1373 |
| Pica8 3290 | 1608 | 4000 |
| IWNetworks 8952 | 1000* | UNK |

**Figure 5 - Flows Handled by Default**

* combination of L3 and L2 flows

The following is the raw data from some of the TCP TTFF testing.

## Sample runs from testing latency of unidirectional flows

| Run 9 | Run 10 | Run 11 | Run 12 | Run 13 | Run 14 | Run 15 |
|---|---|---|---|---|---|---|
| 2.6 | 2.9 | 44.2 | 20.7 | 21.8 | 2.5 | 1.9 |
| 15.6 | 2.6 | 45.3 | 21 | 22.4 | 3.9 | 3.3 |
| 21 | 4.2 | 46.4 | 22.6 | 22.6 | 5.2 | 5.7 |
| 21.2 | 5.5 | 46.7 | 22.3 | 22.1 | 6.5 | 7.5 |
| 21.7 | 9.5 | 48.4 | 23.9 | 23.9 | 8.7 | 8.8 |
| 22 | 14.5 | 49.5 | 24.2 | 24.2 | 11.7 | 10.1 |
| 23 | 16 | 51.7 | 25.6 | 26.5 | 13.9 | 12.3 |
| 23.3 | 17.3 | 52.8 | 25.8 | 26.8 | 16.1 | 14.7 |
| 24.7 | 18.6 | 54.7 | 27.1 | 27 | 18.2 | 15 |
| 25 | 20 | 55.8 | 27.5 | 27.3 | 19.6 | 17.2 |
| 26.3 | 21.3 | 58 | 28.8 | 28.7 | 24.2 | 18.5 |
| 26.6 | 21.6 | 59.1 | 29 | 28.9 | 24.4 | 20.6 |
| 27.9 | 22.9 | 60.5 | 30.4 | 31.2 | 25.8 | 26.9 |
| 28.1 | 23.2 | 60.8 | 30.7 | 31.7 | 26.3 | 28.5 |
| 29.5 | 24.6 | 62.1 | 32 | 31.5 | 26.5 | 28.8 |
| 29.8 | 24.8 | 62.4 | 32.3 | 32 | 26 | 28.2 |

**Figure 6 - Latency of unidirectional flows**