Configuring and Controlling a Pica8 Switch with Ryu

Jason Loux, Temple University
Mentor: Dr. Jie Wu

Introduction

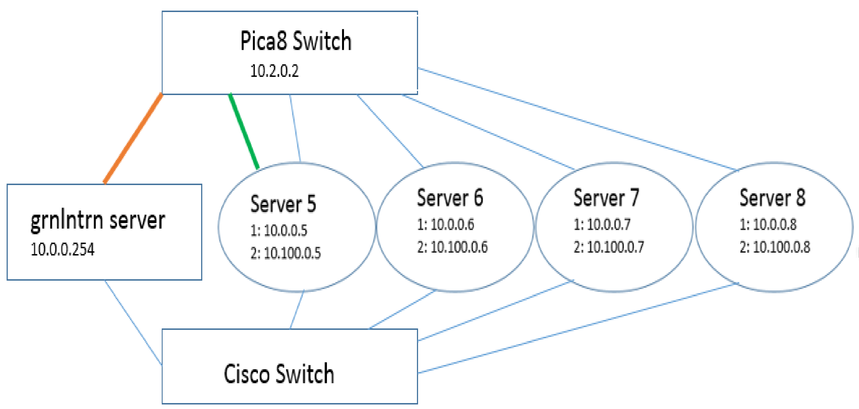
- Current network protocols are exceedingly strict, making the administrative duties of network providers difficult.
- Software Defined Networking (SDN) is a concept which allows control over the operations of a switch via remote software applications.
- Using a Pica8 3297 OpenFlow switch in conjunction with five PowerEdge R210 severs and a Cisco switch, we analyze the response time and the control that the switch maintains while controlled by the Ryu SDN framework.

Objectives

- See how quickly the switch responds to the controller addition of a flow.
- Analyze this time's impact when transferring data to and from servers.
- Exhibit bandwidth control of specific ports using Ryu applications.

Configuration

- A second interface was created on the Pica8 switch so that the server could directly connect to the switch.
- A second interface was configured on each of the four interconnected servers.
- Now an isolated test environment ran directly through Pica8, thereby changing network route from Cisco Switch.



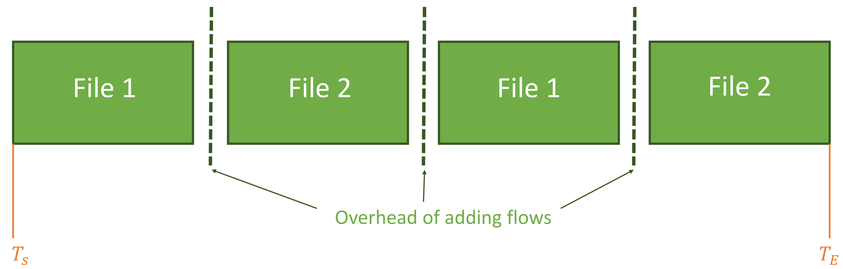
Procedures

Adding a Flow via Ryu Controller Application

- Server5 (sending server) listens for connection to Server6.
- Ryu controller application will then be activated, followed by the client code on Server6.
- Then a flow will be installed and timestamped by the controller application.

Parallel Packet Processing Control

- **T**wo servers take turns sending a large file to one receiving server in the network.
- These times will be compared to the times that are recorded where two files are sent and retrieved simultaneously to the single server.

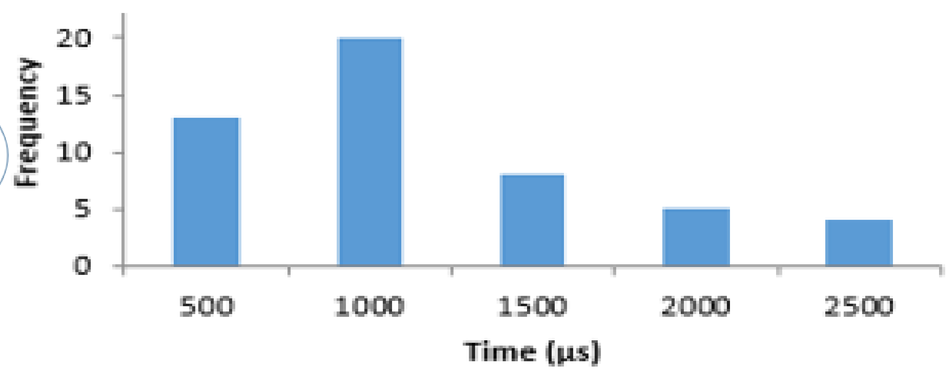


- Next, flows are installed from both Server5 and Server7 to Server6 at the same time, and thus Server6 processes the incoming data in parallel.

Bandwidth Control Using Flow-Queueing

- Bandwidth on Server7 (Port3) will be controlled by first having it run at ¼ port speed and then after x time running at full speed.
- Then we monitor this control through wireshark (network protocol analyzer) to ensure proper speed control.

Flow Response



Parallel Traffic Results

File1	File2	Both[NP]	Both[P]
20368	21233	23098	21473

Time in Milliseconds

- File1 and File2 were measured in the turn-by-turn algorithm for transferring data.
- Both[NP] refers to the non-parallel, turn-by-turn algorithm.
- Both[P] refers to the parallel-processing of data.

Bandwidth Results

```
[  5] local 10.100.0.7 port 5001 connected with 10.100.0.6 port 56386
[  5]  0.0-10.2 sec   315 MBytes   260 Mbits/sec
[  4] local 10.100.0.7 port 5001 connected with 10.100.0.6 port 56387
[  4]  0.0-10.0 sec  1.10 GBytes   941 Mbits/sec
```

Source	Destination	Protocol	Length
10.100.0.6	10.100.0.5	TCP	66
10.100.0.5	10.100.0.6	TCP	17442
10.100.0.6	10.100.0.5	TCP	66
10.100.0.7	10.100.0.6	TCP	1514

Source	Destination	Protocol	Length
10.100.0.7	10.100.0.6	TCP	1514
10.100.0.7	10.100.0.6	TCP	17442
10.100.0.6	10.100.0.7	TCP	66
10.100.0.7	10.100.0.6	TCP	1514
10.100.0.7	10.100.0.6	TCP	8754
10.100.0.7	10.100.0.6	TCP	10202
10.100.0.6	10.100.0.7	TCP	66
10.100.0.6	10.100.0.7	TCP	66
10.100.0.7	10.100.0.6	TCP	5858
10.100.0.6	10.100.0.7	TCP	66
10.100.0.7	10.100.0.6	TCP	14546

Conclusions

- Millisecond flow installation is an expensive amount of time.
- Traffic results showed that the switch is well equipped for large traffic through a single port.
- Bandwidth control was cleanly executed by Ryu SDN framework.