Andrew Katsanevas

CS 4480

4/22/2017

PA3 Assignment Report

Up to this point, I have used Java for all of my networking programming, so for this assignment, I had to take some time to become more familiar with Python syntax. After learning enough about Python to function, I went through all the linked tutorials to understand Mininet and RYU. Read my previous progress report for more information on that.

To code my controller for this assignment, I used the example file simple_switch_13.py given in the app folder as a starting point. Most of the changes I made were in the _packet_in_handler method to handle the arrival of packets. Instead of just receiving packets and sending them along, I modified the controller to create flows when an ARP packet is received. This is because an ARP request comes before any communication between hosts. Once an ARP is received and the flows are created, there is no need for the _packet_in_handler method to worry about new packets coming from the same host.

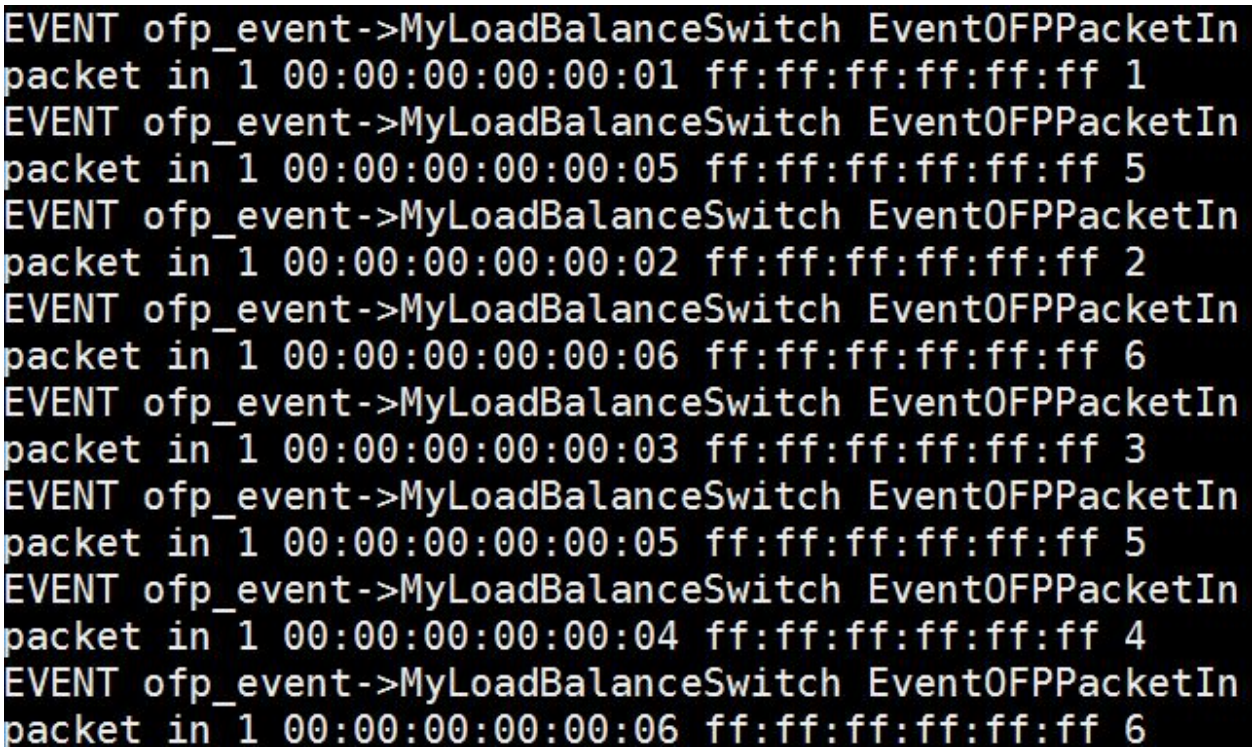
To do this, I first make sure that the packet I'm dealing with is an ARP being sent to s1. If it is, I then retreive the source information from it (the IP and MAC). From there, I set the destination IP/MAC and out_port to whichever server I want (h5 or h6). I make this decision based on my switchBoolean, which essentially flips each time an ARP is sent. This assures that servers are assigned in alternating order, thus balancing. Once the destination host is chosen, I add two flows with new actions and matches. The first flow is the flow from the client to the server, and the second is the server to the client. Once I set up these flows in both directions, I send an ARP back to the client using a helper method send_arp. This helper constructs a new

ARP packet using the appropriate source and destination IPs and MACs, along with a datapath and outport, and sends it.

Screenshots of my program working are below:

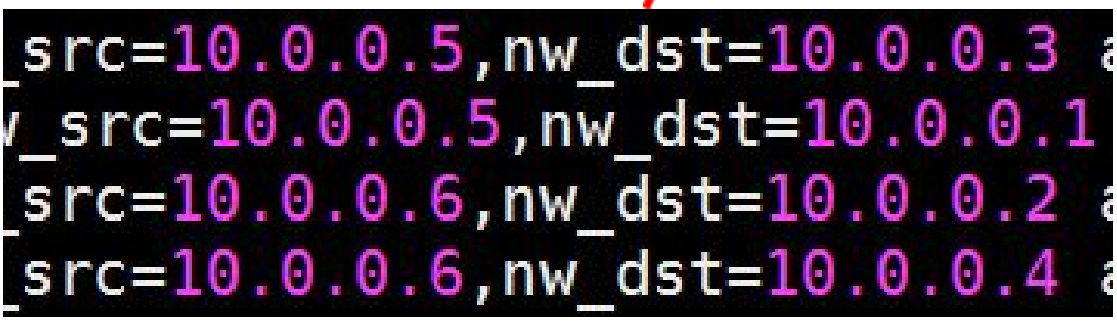
```
mininet> h1 ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=13.6 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.129 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.170 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=0.191 ms
64 bytes from 10.0.0.10: icmp_seq=5 ttl=64 time=0.192 ms
64 bytes from 10.0.0.10: icmp_seq=6 ttl=64 time=0.148 ms
64 bytes from 10.0.0.10: icmp_seq=7 ttl=64 time=0.181 ms
64 bytes from 10.0.0.10: icmp_seq=8 ttl=64 time=0.185 ms
64 bytes from 10.0.0.10: icmp_seq=9 ttl=64 time=0.183 ms
64 bytes from 10.0.0.10: icmp_seq=10 ttl=64 time=0.185 ms
^C
--- 10.0.0.10 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8999ms
rtt min/avg/max/mdev = 0.129/1.524/13.679/4.051 ms
mininet> h2 ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=16.0 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.170 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.175 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=0.173 ms
64 bytes from 10.0.0.10: icmp_seq=5 ttl=64 time=0.164 ms
64 bytes from 10.0.0.10: icmp_seq=6 ttl=64 time=0.142 ms
64 bytes from 10.0.0.10: icmp_seq=7 ttl=64 time=0.124 ms
64 bytes from 10.0.0.10: icmp_seq=8 ttl=64 time=0.073 ms
64 bytes from 10.0.0.10: icmp_seq=9 ttl=64 time=0.131 ms
^C
--- 10.0.0.10 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8000ms
rtt min/avg/max/mdev = 0.073/1.909/16.031/4.992 ms
mininet> h3 ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=12.9 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.146 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.166 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=0.183 ms
64 bytes from 10.0.0.10: icmp_seq=5 ttl=64 time=0.160 ms
64 bytes from 10.0.0.10: icmp_seq=6 ttl=64 time=0.141 ms
64 bytes from 10.0.0.10: icmp_seq=7 ttl=64 time=0.187 ms
64 bytes from 10.0.0.10: icmp_seq=8 ttl=64 time=0.178 ms
64 bytes from 10.0.0.10: icmp_seq=9 ttl=64 time=0.176 ms
64 bytes from 10.0.0.10: icmp_seq=10 ttl=64 time=0.176 ms
^C
--- 10.0.0.10 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9002ms
rtt min/avg/max/mdev = 0.141/1.451/12.999/3.849 ms
mininet> h4 ping 10.0.0.10
```

The functioning pings from each client to s1



The packet receiving events

```
node-0:~> sudo ovs-ofctl -O OpenFlow13 dump-flows s1
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=18.218s, table=0, n_packets=1, n_bytes=42, priority=1,in_port=4,dl_dst=00:00:00:00:00:06 actions=output:6
 cookie=0x0, duration=47.151s, table=0, n_packets=1, n_bytes=42, priority=1,in_port=3,dl_dst=00:00:00:00:00:05 actions=output:5
 cookie=0x0, duration=115.428s, table=0, n_packets=1, n_bytes=42, priority=1,in_port=1,dl_dst=00:00:00:00:00:05 actions=output:5
 cookie=0x0, duration=82.291s, table=0, n_packets=1, n_bytes=42, priority=1,in_port=2,dl_dst=00:00:00:00:00:06 actions=output:6
 cookie=0x0, duration=47.161s, table=0, n_packets=6, n_bytes=588, priority=2,ip,in_port=5,nw_src=10.0.0.5,nw_dst=10.0.0.3 actions=set_field:10.0.0.10->ip_src,output:3
 cookie=0x0, duration=115.441s, table=0, n_packets=8, n_bytes=784, priority=2,ip,in_port=5,nw_src=10.0.0.5,nw_dst=10.0.0.1 actions=set_field:10.0.0.10->ip_src,output:1
 cookie=0x0, duration=82.303s, table=0, n_packets=8, n_bytes=784, priority=2,ip,in_port=6,nw_src=10.0.0.6,nw_dst=10.0.0.2 actions=set_field:10.0.0.10->ip_src,output:2
 cookie=0x0, duration=18.235s, table=0, n_packets=8, n_bytes=784, priority=2,ip,in_port=6,nw_src=10.0.0.6,nw_dst=10.0.0.4 actions=set_field:10.0.0.10->ip_src,output:4
 cookie=0x0, duration=47.162s, table=0, n_packets=7, n_bytes=630, priority=2,ip,in_port=3,nw_dst=10.0.0.10 actions=set_field:10.0.0.5->ip_dst,output:5
 cookie=0x0, duration=18.241s, table=0, n_packets=9, n_bytes=826, priority=2,ip,in_port=4,nw_dst=10.0.0.10 actions=set_field:10.0.0.6->ip_dst,output:6
 cookie=0x0, duration=115.445s, table=0, n_packets=9, n_bytes=826, priority=2,ip,in_port=1,nw_dst=10.0.0.10 actions=set_field:10.0.0.5->ip_dst,output:5
 cookie=0x0, duration=82.304s, table=0, n_packets=9, n_bytes=826, priority=2,ip,in_port=2,nw_dst=10.0.0.10 actions=set_field:10.0.0.6->ip_dst,output:6
```



The flow table shows the correct distribution