CS 39006: Networks Lab
Assignment 5: Implementation of L2 Forwarding and L3 Static Routing Protocol

Date: 07th March, 2017

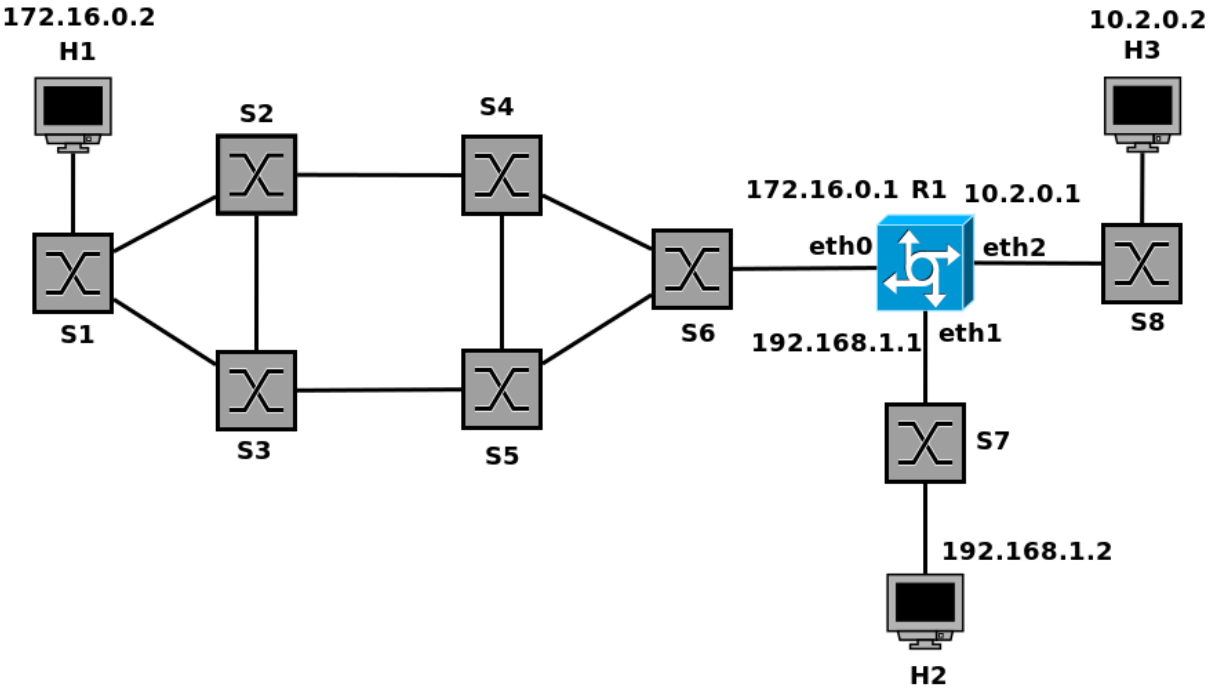
Objective:

The objective of this assignment is to understand L2 Forwarding and L3 Routing mechanism using custom topologies in Mininet. The corresponding L2 forwarding and L3 routing protocols will be implemented as a part of the POX controller in Mininet.

Background:

To give a brief background, here is an example of how a data packet is forarded from one host to another via L2 switches and L3 routers.

Consider the following topology.



Here, H1, H2 and H3 are end hosts, S1 – S8 are the L2 switches and R1 is a L3 router. A packet transfer from H1 to H3 has following steps.

Pr-configurations:

1. H1: Default Gateway – R1-eth0 (172.16.0.1)
2. R1 has a routing table. We use a very simplified static routing table at R1 as follows.

Destination	Next Hop	Interface
172.16.0.2	172.16.0.1	R1-eth0
10.2.0.2	10.2.0.2	R1-eth2
192.168.1.2	192.168.1.2	R1-eth1

Think of why we keep both the next hop and interface information!

Steps for L2 Forwarding and L3 Routing:

- H1 constructs the IP packet and sets following fields in the IP header:
  - Source IP: 172.16.0.2
  - Destination IP: 10.2.0.2
- H1 uses default gateway as its L3 next hop. So its next hop would be 172.16.0.1
- It looks up the MAC table for the MAC address of 172.16.0.1. If the MAC address is there, it uses that MAC address. Otherwise it uses the ARP protocol to get the MAC address of 172.16.0.1
  - H1 broadcasts a ARP request corresponding to 172.16.0.1
  - Once R1-eth0 receives that ARP request, it sends back a ARP reply with its own MAC address
- H1 then constructs the MAC header as follows:
  - Source MAC: MAC of H1
  - Destination MAC: MAC of R1-eth0 (corresponding to 172.16.0.1)
- The packet is then forwarded to L2. It then uses the Spanning Tree Protocol (STP) to broadcast the packet at L2.
- At L2, following the STP, every switch S1, S2, S3, S4, S5, S6 as well as R1-eth0 receives the packet. At every hop, the L2 device extracts the destination MAC address from the MAC header, and matches it with its own MAC address. Only R1-eth0 gets a match, and so it accepts the packet. R1-eth0 forwards the packet for IP processing at R1.
- R1 looks up the destination IP in the packet header. Then it makes a destination look-up in the IP routing table. The routing table says that the next hop is 10.2.0.2 for the destination 10.2.0.2, and the interface for communication is R1-eth2. So this next hop and interface information is forwarded to the MAC layer of R1.
- R1 again uses the MAC lookup (based on ARP) for 10.2.0.2. It constructs the MAC header as follows,
  - Source MAC: MAC for R1-eth2
  - Destination MAC: MAC of H2
- The packet then again follows the L2 forwarding to eventually reach at H2.

We'll now implement this L2 forwarding and L3 routing as a part of the Mininet POX controller.

Assignment Statement:

Exend the POX controller to implement the L2 forwarding and L3 routing mechanisms. To implement L3 routing, you can consider a simplified static routing table as given in the example. Essentially your assignment will have three parts.

1. **ARP Implementation –** Use ARP to find out the MAC address of the L3 next hop.
2. **L2 Forwarding** - As L3 next hop may not have direct connection, and there can be a mesh of L2 switches in between the current node and its L3 next hop, use STP based broadcasting (L2 forwarding) to forward the packet to the next hop.
3. **IP Forwarding –** At the intermediate L3 routers, look up the routing table to find out the next hop and the interface information to reach that next hop.

Therefore, you need to create two python classes by extending the POX controller – *l2_switch()* which will be attached with every L2 switches to handle L2 forwarding and *l3_switch()* which will be attached to the routers to handle the ARP lookup and IP forwarding.

As network is a event driven system, you need to understand how *events* are generated in POX and how you can add a *Listener* corresponds to an event. The important events that you may need to look into are as follows:

a) *event.connection*
b) *of.ofp_packet_out, of.ofp_packet_in, of.ofp_action_output*
c) *Core.openflow.addListeners*
d) *core.getLogger*

you can check this tutorial to get an idea about how you can use this events and write your own event listener: https://openflow.stanford.edu/display/ONL/POX+Wiki

Further, you have to handle various packet headers in your code. For that, you need to llok into the *pox.lib.packet* from the POX library. A sample example of handling ARP packets are as follows.

```
def _handle_PacketIn (self, event):
       packet = event.parsed
       if packet.type == packet.ARP_TYPE:
              if packet.payload.opcode == arp.REQUEST:
                     arp_reply = arp()
                     arp_reply.hwsrc = <requested mac address>
                     arp_reply.hwdst = packet.src
                     arp_reply.opcode = arp.REPLY
                     arp_reply.protosrc = <IP of requested mac-associated machine>
                     arp_reply.protodst = packet.payload.protosrc
                     ether = ethernet()
                     ether.type = ethernet.ARP_TYPE
                     ether.dst = packet.src
                     ether.src = <requested mac address>
                     ether.payload = arp_reply
                     #send this packet to the switch
              elif packet.payload.opcode == arp.REPLY:
                     print "It's a reply; do something cool"
              else:
                     print "Some other ARP opcode, probably do something smart here"
```

You can also look into this tutorial for learning about the POX controller:
https://pypi.python.org/pypi/pox

For this implementation, use a topology similart to the one as discussed in the earlier figure.

Submission Instructions:

You need to prepare a report that will contain the followings.

1. Steps followed in executing the experiments.
2. Observations from the experiments.
3. The scripts corresponding to topology construction, controller, L2 forwarding and L3 routing.
4. Intuitive justification behind the observations.

You need to submit the report and relevant scripts (source files) in a single compressed (tar.gz) file. Rename the compressed file as Assignment_4_Roll1_Roll2.tar.gz, where Roll1 and Roll2 are the roll numbers of the two members in the group. Submit the compressed file through Moodle by the submission deadline. The submission deadline is: **March 14, 2017 02:00 PM.** Please note that this is a strict deadline and no extension will be granted.

Please note that your submission will be awarded zero marks without further consideration, if it is find to be copied. In such cases, all the submissions will be treated equally, without any discrimination to figure out who has copied from whom.