Assignment N0.2

Aim: After studying open source POX and Floodlight controller, Install controller and run custom topology using remote controller like POX and floodlight controller. Recognize inserted flows by controllers.

Objective:

1. Learn installation of controller
2. run custom topology using remote controller like POX and floodlight controller

Theory:

open source POX

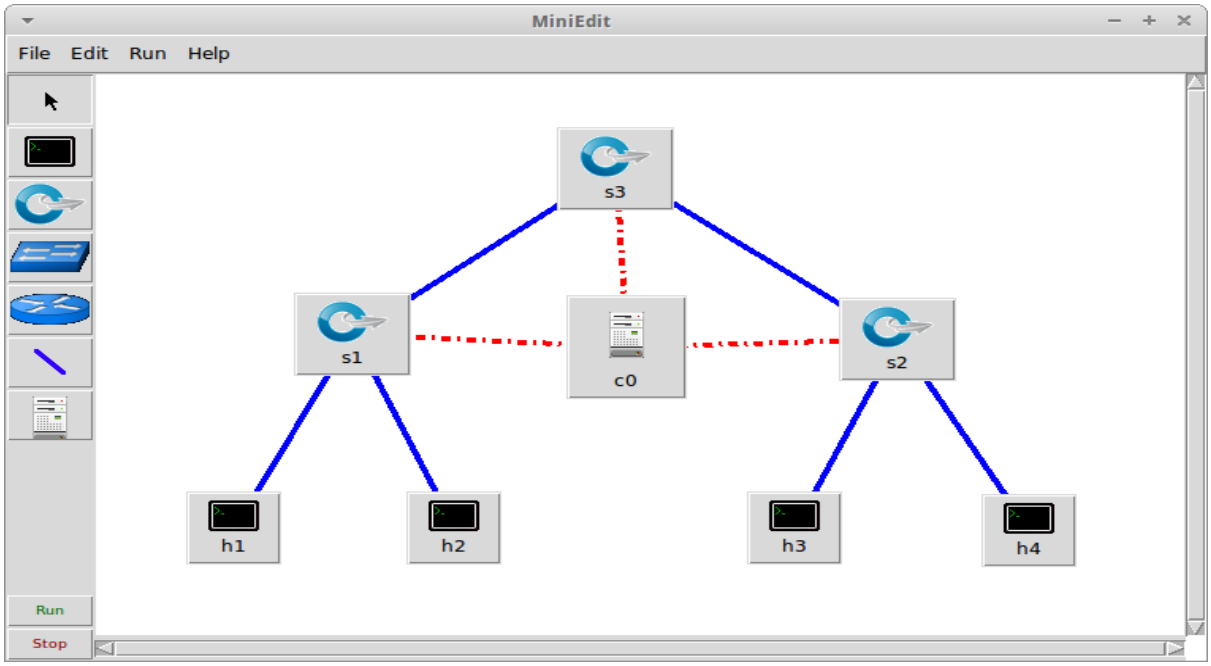
POX provides a framework for communicating with SDN switches using either the OpenFlow or OVSDB protocol. Developers can use POX to create an SDN controller using the Python programming language. It is a popular tool for teaching about and researching software defined networks and network applications programming.
POX can be immediately used as a basic SDN controller by using the stock components that come bundled with it.
Developers may create a more complex SDN controller by creating new POX components. Or, developers may write network applications that address POX's API

Floodlight controller :

Floodlight Controller is an SDN Controller developed by an open community of developers, many of which from Big Switch Networks, that uses with the OpenFlow protocol to orchestrate traffic flows in a software-defined networking (SDN) environment. OpenFlow is one of the first and most widely used SDN standards; it defines the open communications protocol in an SDN environment that allows the SDN Controller (brains of the network) to speak to the forwarding plane (switches, routers, etc.) to make changes to the network.

Build the network and use a remote controller

Build the network consisting of a tree to switches with a central core switch connected to two other switches that are connected to two hosts, each. Connect a controller to all the switches.



Start the POX controller

Before we start the POX SDN controller, we need to determine which components we want to run when we start the controller.

Select the POX components to run

To select the correct stock components, determine what behavior we want the network of switches to exhibit. Then we will select the stock POX component that provides that functionality.

In this practical, we will use components that make POX work like a Layer 2 learning switch, and that dump copies of packets received by the controller to the controller's log file (so we can see what packets the controller sees), and that list controller events to the console log screen in an easy-to-read format. According to the POX documentation, the stock components that do these tasks are: forwarding.l2_pairs, info.packet_dump, samples.pretty_log, and log.level.

Start POX

To start the POX controller with the selected stock components, enter the following command on a terminal session connected to the Mininet VM.

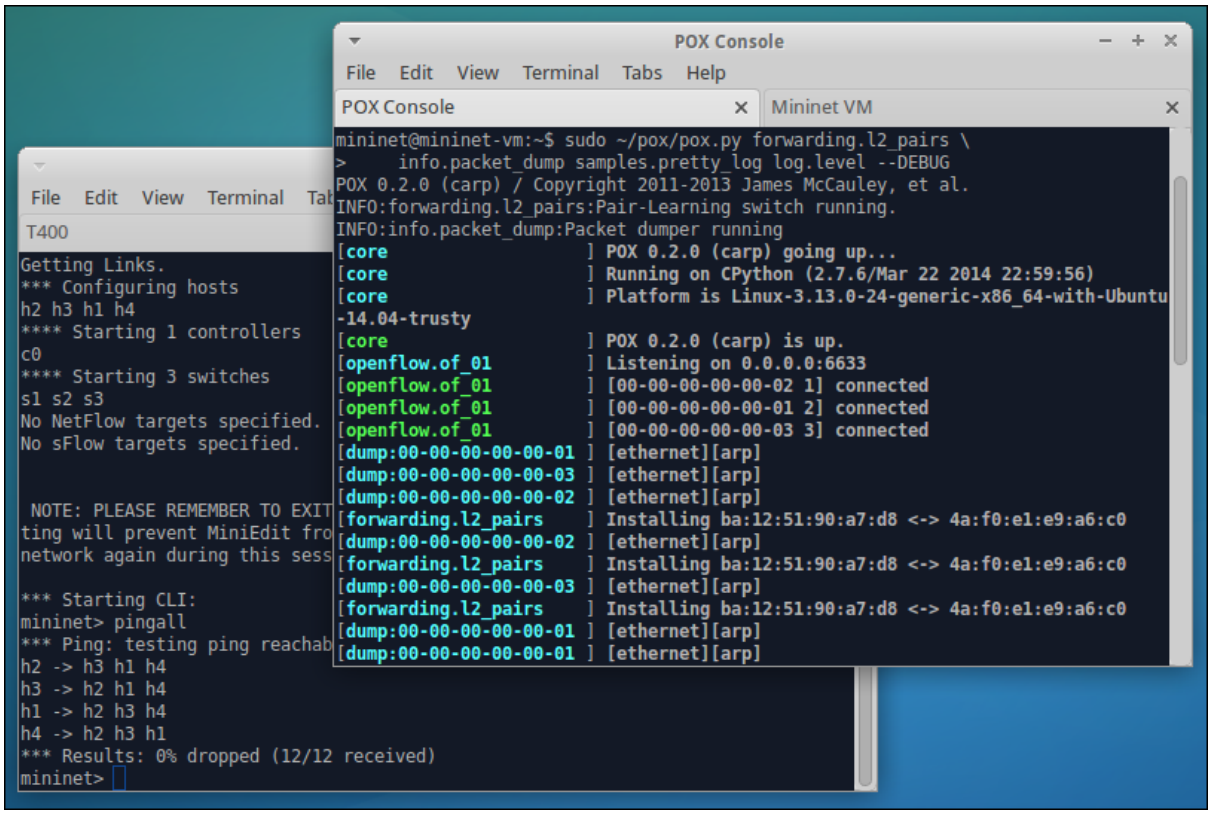
```
mininet-vm:~$ sudo ~/pox/pox.py forwarding.l2_pairs \

info.packet_dump samples.pretty_log log.level --DEBUG
```

In the POX console, see that the log shows the controller starts and connects to the switches previously set up by the Mininet network simulator:

Test the controller

The forwarding.l2_pairs component is a very simple application that just matches MAC addresses so it creates a simple scenario to study.

Generate some test traffic between hosts to see how POX builds flows in the network. Run the Mininet `pingall` command, which runs ping tests between each host in the emulated network. This generates traffic to the controller every time a a switch receives a packet that has a destination MAC address that is not already in its flow table.



You can see in the POX console window the log messages showing what is happening. When the POX controller running the forwarding.l2_pairs component receives a packet from a switch, it tells the switch to flood the ARP packet out its other ports to other switches or hosts. One host eventually responds to the ARP request and then the forwarding.l2_pairs component sends OpenFlow messages to each switch to load the required flows into the switch flow tables.

Checking flow tables

To see the contents of the flow tables on all switches, execute the Mininet command:

```
mininet> dpctl dump-flows
```

To check ARP tables on each host, execute the Mininet arp command. For example, to show the ARP table for host h1, enter the following command:

```
mininet> h1 arp
```

To clear all flow tables on all switches, enter the Mininet command:

```
mininet> dpctl del-flows
```

Conclusion

We learn how to install controller and run custom topology using remote controller like POX and floodlight controller. Recognize inserted flows by controllers