CS 640 Introduction to Computer Networks
Lab 4 Overview

Posted on Canvas

Due: Friday Dec 2 2022 11:59PM

Agenda

1. Objectives
2. Implementation
3. Short example & Sample output
4. Grading Rubrics

Objectives

1. Contrast SDN applications and traditional network control planes
2. Create SDN applications that use proactive or reactive flow installation
    1. Layer-3 Routing Application
    2. Distributed Load Balancer Application

Implementation

1. Setup
    1. **Install the required mininet version(see description), and the required packages.**
    2. Download, Install and Patch floodlight-plus package.
    3. Build and run FloodlightWithApps.jar with provided prop file as args.
    4. Start mininet for a specific topology
    5. Test the setup on a provided topology by pinging two hosts.
2. Code
    1. In Part 2 – Modify `L3Routing.java` in `edu.wisc.cs.sdn.apps.l3routing`
    2. In Part 3 – Modify `LoadBalancer.java` in ‘`edu.wisc.cs.sdn.apps.loadbalancer`’
3. Test
    1. Use the provided topologies to test your implementation (See Rubric)
    2. Test L3 Routing after implementing Load Balancer
        - Due to some unknown issues in floodlight-plus, Hosts in L3 Routing do not get added without the Load Balancer.

Layer-3 routing

1. **TODO**

   install and remove flow table entries from SDN switches such that traffic is forwarded to a host using the shortest path.

2. Implementation

1. Extract link information using getLinks(), and check the related functions in Class Link. You should be able to get enough information to start the shortest path algorithm.
2. Computing shortest paths(Bellman-Ford or other pair shortest path algorithms).

   For example, for each different destination switch, you can calculate the

   <source switch, port of source switch to forward packet along shortest path>
3. For every host, you should install rules at each switch to enable the packets to be forwarded using SwitchCommands.installRule()

   You may need to review the following classes: OFAction, OFInstruction, SwitchCommands

3. Test

1. Use the provided topologies to test your implementation (See Rubric 1-10)
2. Test L3 Routing after implementing Load Balancer
3. Use commands to test such as link s1 s2 down **(lab4 description p. 13-14)**

Load balancing

1. TODO & Implementation suggestions

receive(): Handle incoming packets sent from switches

1. Send an ARP reply for ARP requests for virtual IPs

   check ARP, Ethernet, getVirtualMAC().Use SwitchCommands.sendPacket() to send.
2. for TCP SYNs sent to a virtual IP, select a host and install connection-specific rules to rewrite IP and MAC addresses

   check IPV4, TCP(cast from IPV4.getPayload()), OFActionSetField.

   Should install two rules for each connection.
3. ignore all other packets

switchAdded(): Install rules to send: rules are added using SwitchCommands.installRule()

1. packets from new connections to each virtual load balancer IP to the controller

   check field *instances*, getMatchCriteriaForIP(), and OFMatch.setNetworkProtocol()
2. ARP packets to the controller

   check OFMatch.setDataLayerType()
3. All other packets to the next rule table in the switch

   Hint: use an appropriate priority and OFMatch without restrictions

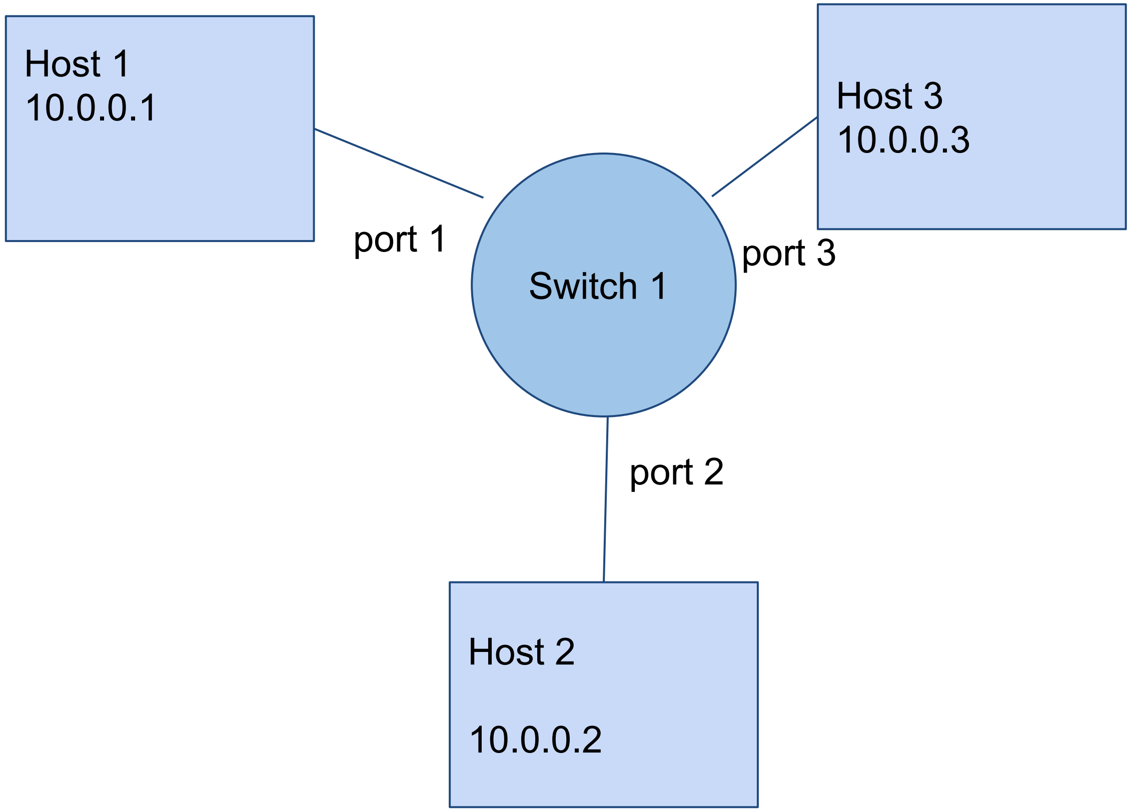
Load balancing

2. Test

1. Use the provided topologies to test your implementation (See Rubric 12-19)
2. Use tcpdump -v -n -i hN-eth0 to check the packets
3. Running the below command should download a page.

```
h1 curl http://<virtualIP>:<port>/index.html
e.g: h1 curl http://10.0.100.1:80/index.html
```

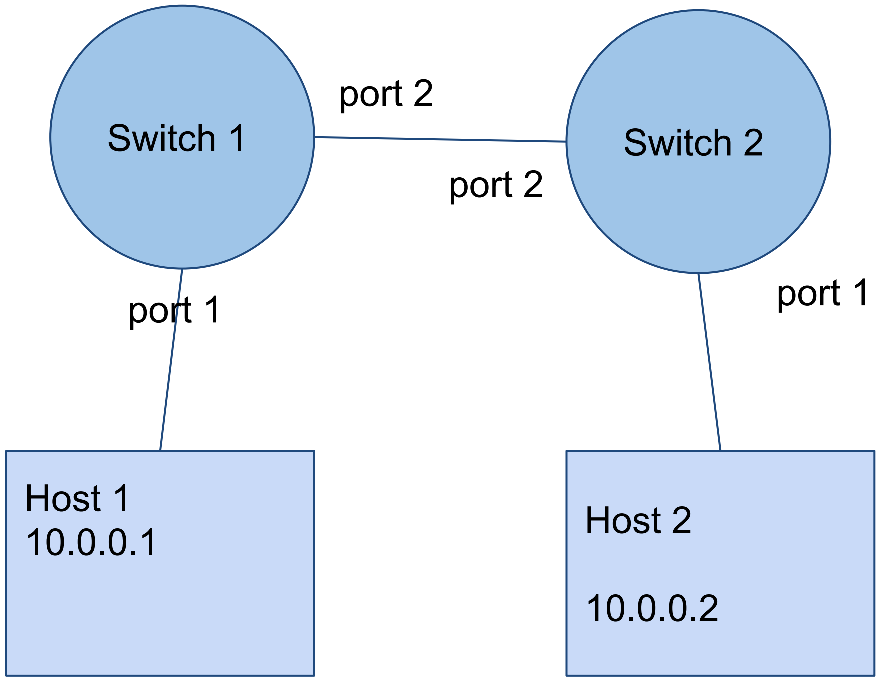
Example: Topology single,3



Example: L3 using Single topo

```
mininet@mininet-vm:~$ sudo ovs-ofctl -O OpenFlow13 dump-flows s1 (contents of an SDN s1’s flow tables)
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=35.243s, table=0, n_packets=0, n_bytes=0, priority=1,tcp,nw_dst=10.0.100.1
actions=CONTROLLER:65509
 cookie=0x0, duration=35.245s, table=0, n_packets=0, n_bytes=0, priority=1,tcp,nw_dst=10.0.110.1
actions=CONTROLLER:65509
 cookie=0x0, duration=35.243s, table=0, n_packets=5, n_bytes=210, priority=1,arp actions=CONTROLLER:65509
 cookie=0x0, duration=35.243s, table=0, n_packets=0, n_bytes=0, priority=0 actions=goto_table:1
 cookie=0x0, duration=34.3s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.2 actions=output:2
 cookie=0x0, duration=33.298s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.3 actions=output:3
 cookie=0x0, duration=34.307s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.1 actions=output:1
```

Example: L3 using Linear,2 topo



Example: L3 using Linear,2 topo

S1:
From terminal: sudo ovs-ofctl -O OpenFlow13 dump-flows s1

```
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=15.458s, table=0, n_packets=0, n_bytes=0, priority=1,tcp,nw_dst=10.0.100.1
actions=CONTROLLER:65509
 cookie=0x0, duration=15.458s, table=0, n_packets=0, n_bytes=0, priority=1,tcp,nw_dst=10.0.110.1
actions=CONTROLLER:65509
 cookie=0x0, duration=15.458s, table=0, n_packets=2, n_bytes=84, priority=1,arp actions=CONTROLLER:65509
 cookie=0x0, duration=15.458s, table=0, n_packets=2, n_bytes=122, priority=0 actions=goto_table:1
 cookie=0x0, duration=14.409s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.2 actions=output:2
 cookie=0x0, duration=15.416s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.1 actions=output:1
```

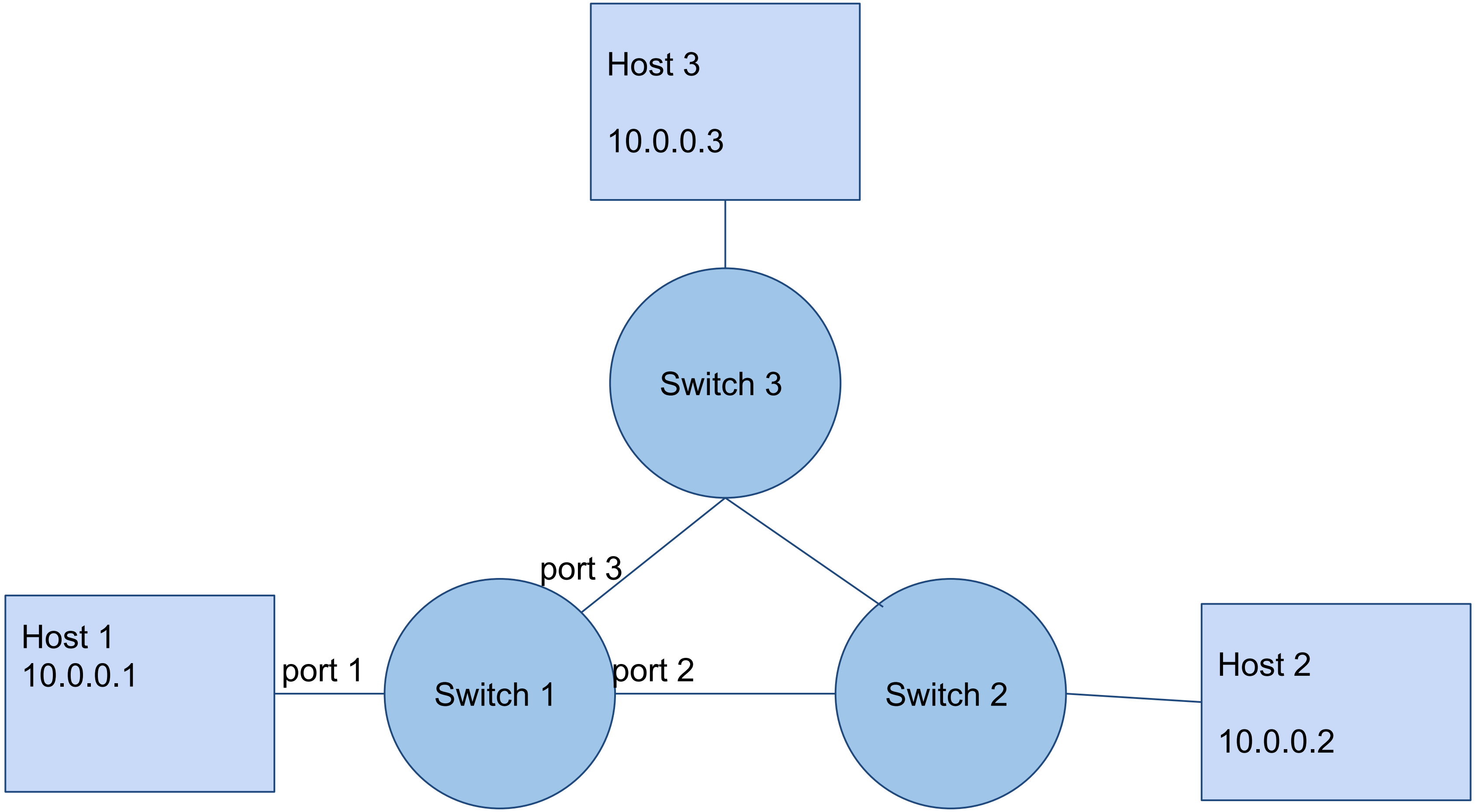
Example: L3 using Linear,2 topo

S2:

From terminal:sudo: ovs-ofctl -O OpenFlow13 dump-flows s2

```
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=30.019s, table=0, n_packets=0, n_bytes=0, priority=1,tcp,nw_dst=10.0.100.1
actions=CONTROLLER:65509
 cookie=0x0, duration=30.031s, table=0, n_packets=0, n_bytes=0, priority=1,tcp,nw_dst=10.0.110.1
actions=CONTROLLER:65509
 cookie=0x0, duration=30.019s, table=0, n_packets=2, n_bytes=84, priority=1,arp actions=CONTROLLER:65509
 cookie=0x0, duration=30.019s, table=0, n_packets=2, n_bytes=122, priority=0 actions=goto_table:1
 cookie=0x0, duration=28.971s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.2 actions=output:1
 cookie=0x0, duration=29.976s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.1 actions=output:2
```

Example: LB using Triangle topo



Example: LB using Triangle topo

S1:
From terminal: sudo ovs-ofctl -O OpenFlow13 dump-flows s1

```
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=49.683s, table=0, n_packets=0, n_bytes=0, priority=1,tcp,nw_dst=10.0.100.1
actions=CONTROLLER:65509
 cookie=0x0, duration=49.685s, table=0, n_packets=0, n_bytes=0, priority=1,tcp,nw_dst=10.0.110.1
actions=CONTROLLER:65509
 cookie=0x0, duration=49.68s, table=0, n_packets=2, n_bytes=84, priority=1,arp actions=CONTROLLER:65509
 cookie=0x0, duration=49.68s, table=0, n_packets=11, n_bytes=679, priority=0 actions=goto_table:1
 cookie=0x0, duration=48.534s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.2 actions=output:2
 cookie=0x0, duration=47.52s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.3 actions=output:3
 cookie=0x0, duration=49.49s, table=1, n_packets=0, n_bytes=0, priority=1,ip,nw_dst=10.0.0.1 actions=output:1
```

Grading Rubrics (Layer 3 Routing)

SI No	Test case	Points	Topo
1	Flow entries for each host matches host's IP	3	single,3
2	Flow entries have correct timeout	3	single,3
3	Packets are forwarded between hosts on the same switch	3	single,3
4	Flow entries are sent to other switch to reach host on other switch	3	linear,2
5	Packets are forwarded between hosts on two different switches	3	linear,2
6	Packets are forwarded between hosts on many different switches	3	linear,5
7	Shortest-paths are computed correctly in a network with loops	3	someloops
8	Paths are recomputed when link goes down	3	someloops
9	Paths are recomputed when link comes up	3	someloops
10	Paths are recomputed correctly when link goes up and/or down	3	someloops
11	Shortest-paths are computed correctly in a mesh	3	mesh,5

Grading Rubrics (Load Balancing)

Sl No	Test case	Points	Topo
12	Every switch has rules to send all other packets to next table	3	triangle
13	Every switch has rules to send TCP packets for virtual IP to controller	3	triangle
14	Every switch has rules to send ARP requests for virtual IP to controller	3	triangle
15	Pingall success for load balancer	3	triangle
16	Rules have idle timeout = 20s	3	single,6
17	SYN for virtual IP results in correct rule for packets from client	3	single,6
18	SYN for virtual IP results in correct rule for packets from server	3	single,6
19	Connections are round-robined between hosts to test load balancing	6	single,6

Keep the Rubric handy during implementation.

Thank You

Use Piazza or Office Hours for any doubts