Programming Assignment 1

Deadline:10/04/2018

Overview

In this assignment, you will program the OpenFlow controller POX and use it to implement two applications.

Task 1: Firewall

In this part, your task is to implement a layer-2 firewall application using the POX controller. Your application is provided with a file containing pairs of MAC addresses that are not allowed to communicate with each other.

When a connection between a switch and a controller is up, the application installs flow entries to disable the traffic between each pair of the MAC addresses in the list. To make things simple, you can install the firewall rules on all switches in the network.

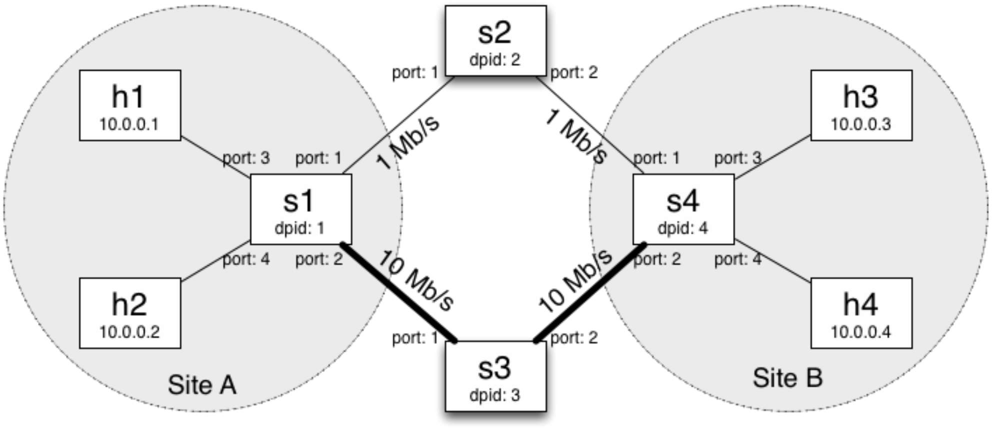
`firewall-policies.csv`: a sample list of MAC address pairs whose end-to-end communication will be disabled by the firewall application.

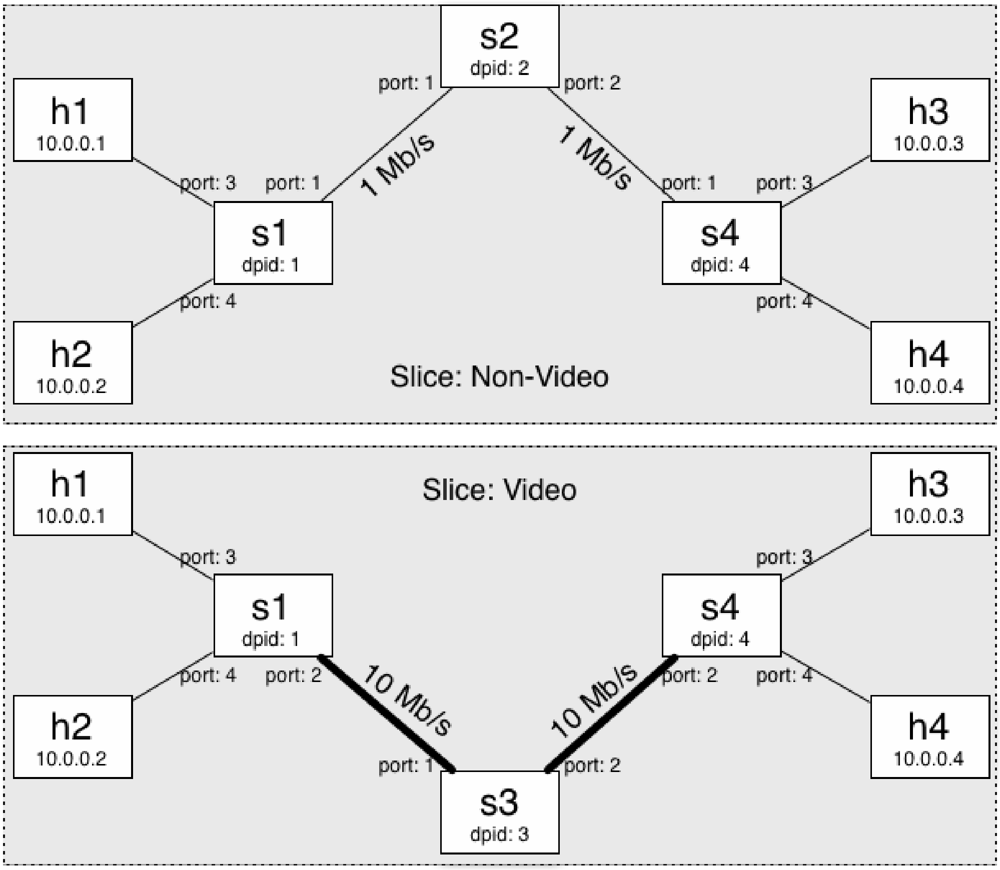
Task 2: Flowspace Slicing

In this part, your task is to slice the network based on the application that is sending the traffic. In principle, SDN networks can be sliced based on any attributes of the flowspace.

The topology is shown below. You need to treat the video and non-video traffic differently. All the video traffic will go through the high bandwidth path (video slice), and all other traffic will go through the low bandwidth path (non-video slice). To simplify this task, we assume that all the video traffic use TCP port 10000.

Topology


s2
dpid: 2
port: 1
port: 2
h1
10.0.0.1
h3
10.0.0.3
1 Mb/s
1 Mb/s
port: 3
port: 1
port: 1
port: 3
s1
dpid: 1
s4
dpid: 4
port: 4
port: 2
port: 2
port: 4
10 Mb/s
10 Mb/s
h2
10.0.0.2
h4
10.0.0.4
port: 1
port: 2
s3
dpid: 3
Site A
Site B


s2
dpid: 2
port: 1
port: 2
h1
10.0.0.1
h3
10.0.0.3
1 Mb/s
1 Mb/s
port: 3
port: 1
port: 1
port: 3
s1
dpid: 1
s4
dpid: 4
port: 4
port: 4
h2
10.0.0.2
Slice: Non-Video
h4
10.0.0.4
h1
10.0.0.1
Slice: Video
h3
10.0.0.3
port: 3
port: 3
s1
dpid: 1
s4
dpid: 4
port: 4
port: 2
port: 2
port: 4
10 Mb/s
10 Mb/s
h2
10.0.0.2
port: 1
port: 2
h4
10.0.0.4
s3
dpid: 3

Handling Conflicts

It is possible to create two applications; POX supports multiple concurrently running applications. However, it is possible that rules, in particular with multiple applications, will conflict with another application. In this assignment, the network slicing application should not allow communication which is disabled by the firewall application. You are to make sure that no conflicting rules are installed by two different applications.

Hint: Such conflict can be handled by setting different priority value to different rules. OpenFlow message structure ofp_flow_mod has an attribute called priority.

Notice also that the topology itself has a loop.

Testing your Code

Once you have your code, copy it to ~/pox/ext directory on your machine.

Run POX controller:

```
$ cd ~/pox

$ ./pox.py %NAME%
```

In another terminal, run the network topology.

First, you can check whether your firewall application is working. In Mininet, try pingall, you should see all hosts can connect to each other except for those which are prevented by the firewall.

Second, you can verify whether you have successfully sliced the network for video and non-video flow. For example, you can test the two paths between h2 and h4 as below:

```
mininet > h4 iperf -s -p 10000 &

mininet > h4 iperf -s -p 22 &

mininet > h2 iperf -c h4 -p 10000 -t 2 -i 1
```

The result should show the bandwidth is about 10.00Mbits/sec

```
mininet > h2 iperf -c h4 -p 22 -t 2 -i 1
```

The result should show the bandwidth is about 1.00Mbit/sec

Submission

Please submit your assignment **BEFORE 23:50 10th April 2018**.
Submissions after the deadline will not be graded.

Attach a submission comment file which includes short description of how to run your code and full names with IDs of your team members. Specify, who from the team is a corresponding member if a correspondence is required.

ZIP all your code, files, and documents (if any) into a single file named: `assignment1_sdn_%Memeber_ID%.zip`
Where %MEMBER_ID% is the ID of the corresponding member.

Submissions in any other name formats will be penalized. You are strongly encouraged to submit your first draft of your assignment as soon as possible. Later (but before the deadline) you can submit the final version of your assignment. Only the last submission will be graded.

Demo on 15th and 16th April

All students are required to give a demo of their assignment. Please choose and fix a proper time slot for you in one of the two days 15th or 16th November 2018.

This script should work exactly as expected to get full grade. Points will be subtracted for missing or incorrect functionality. The penalty for late submission is 10% per day. The submitted code will be tested for plagiarism using. Any attempt to plagiarize will be accordingly punished.

Before contact, please make sure that you have formulated your question clearly and that you have already studied the POX wiki, the Mininet documentation, and the OpenFlow tutorial thoroughly.

Good luck!