This is the test document for the final project. It includes two parts. One for Layer-3 “Shortest-Path Switching” Routing Application Implementation. Another for Distributed Load Balance Routing Application Implementation.

**Part 3:**

**Layer-3 “Shortest-Path Switching” Routing Application Implementation**

I first compile the application by typing *ant* under the project3 directory, and I will get the FloodlightWithApps.jar file.

I can then test the application with the command.

***Step 1. Run the application with command***

*java -jar FloodlightWithApps.jar -cf l3routing.prop*

**IMPORTANT**: This part written in red should be paid great attention. You should refactor the l3routing.prop accordingly before running test. **I also provide the l3routing.prop in the source code.** **I recommend that you just use the file provided by me.**

It should be noticed that the content of l3routing.prop needs to be changed. The original l3routing.prop calls the l3Routing package which I do not use for this project.

I refactor the loadbalancer.prop by changing from edu.nyu.cs.sdn.apps.sps.L3Routing and edu.nyu.cs.sdn.apps.sps.L3Routing.table=0 to edu.nyu.cs.sdn.apps.sps.ShortestPathSwitching and edu.nyu.cs.sdn.apps.sps.ShortestPathSwitching.table = 0

***Step 2. Open another terminal and run the mininet***

*sudo python run\_mininet.py mesh,5*

***Step 3. Test that all the hosts are connected together in mininet terminal***

*pingall*

***Step 4. The program provides a function that allows you to monitor the current shortest path logData(), when it’s activated, you can view the shortest path table.***

文本

描述已自动生成

***Step. 5 You can also view the flow table by opening another terminal and run***

*sudo ovs-ofctl -O OpenFlow13 dump-flows s1*

***Step.6 Testing to modify the links, switches, and hosts***

*link s1 s3 down*

*link s1 s3 up*

*link s1 h1 down*

*link s1 h1 up*

The above commands are running in the mininet CLI.

*sudo ovs-vsctl del-br s1*

This command is running in another terminal.

**Part 4:**

**Distributed Load Balance Routing Application Implementation**

I first compile the application by typing *ant* under the project3 directory, and I will get the FloodlightWithApps.jar file.

I can then test the application with the command.

***Step 1. Run the application with command***

*java -jar FloodlightWithApps.jar -cf loadbalancer.prop*

**IMPORTANT**: This part written in red should be paid great attention. You should refactor the loadbalancer.prop accordingly before running test. **I also provide the loadbalancer.prop in the source code.** **I recommend that you just use the file provided by me.**

Like Part 3, the content of loadbalancer.prop needs to be changed. The original loadbalancer.prop calls the l3Routing package which I do not use for this project.

I refactor the loadbalancer.prop by changing from edu.nyu.cs.sdn.apps.l3routing.L3Routing and edu.nyu.cs.sdn.apps.l3routing.L3Routing.table=1 to edu.nyu.cs.sdn.apps.sps.ShortestPathSwitching and edu.nyu.cs.sdn.apps.sps.ShortestPathSwitching.table=1, respectively

***Step 2. Open another terminal and run the mininet***

*sudo python run\_mininet.py assign1*

***Step 3. Test that all the hosts are connected together in mininet terminal***

*pingall*

***Step. 4 You can also view the flow table by opening another terminal and run***

*sudo ovs-ofctl -O OpenFlow13 dump-flows sN (where N is the number of a switch)*

***Step.5 Testing to ping and issue web requests to virtual IPs.***

In mininet run

*h1 ping -c 1 10.0.100.1*

*h1 curl 10.0.100.1*

I provide couple of functions with which you can view the content of the ARP reply packets and what IP and MAC address does virtual IPs rewrite.