Contents

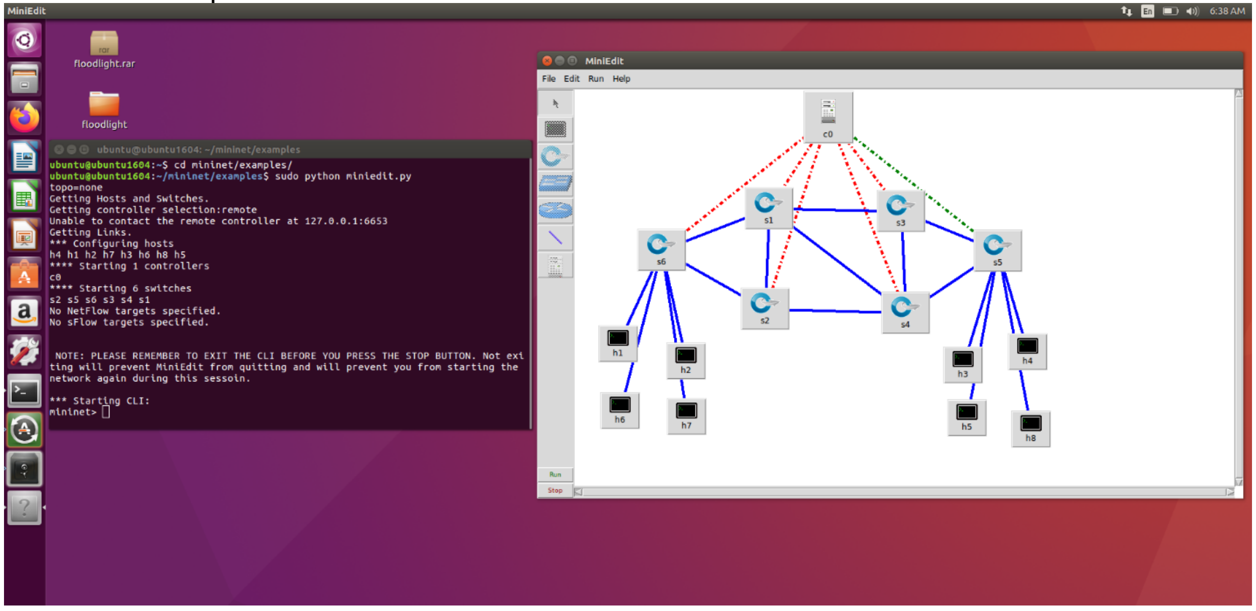
1. Introduction

The purpose of this report is to analyse and examine a network infrastructure using SDN core concepts and technologies. This will include a proof-of-concept design using an open source SDN solution known as Mininet, as well as an overview of the Floodlight Openflow controller, network ping tests, packet analysis with Wireshark, firewall implementation, load balancing and network monitoring.

2. Initial Network Configuration

2.1. Proof of Concept

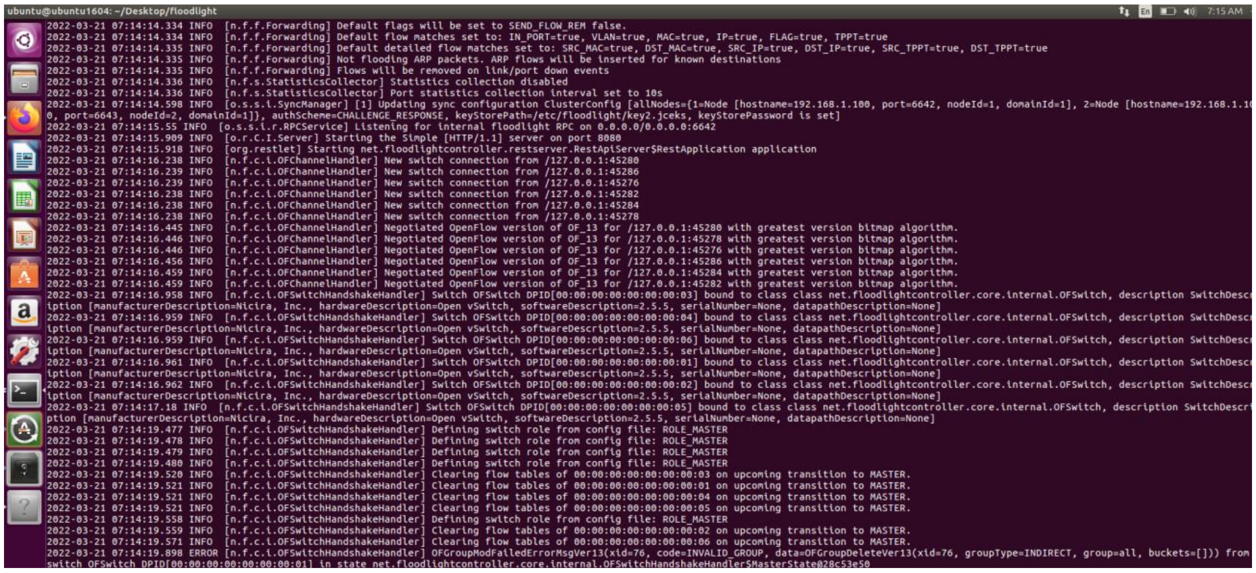
The topology was created via Miniedit, accessed through a command line python script as shown in figure 1. The "c0" represents the central network floodlight controller, which is connected to 6 switches, delegating traffic between 8 different hosts machines. Additional options that were required to be changed were the controller port to 6653, the controller type to remote and the protocol to TCP.



```
ubuntu@ubuntu1604:~$ cd mininet/examples/
ubuntu@ubuntu1604:~/mininet/examples$ sudo python miniedit.py
topo=none
Getting Hosts and Switches.
Getting controller selection:remote
Unable to contact the remote controller at 127.0.0.1:6653
Getting Links.
*** Configuring hosts
h4 h1 h2 h7 h3 h6 h8 h5
**** Starting 1 controllers
c0
**** Starting 6 switches
s2 s5 s6 s3 s4 s1
No NetFlow targets specified.
No sFlow targets specified.


 NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exi
ting will prevent MiniEdit from quitting and will prevent you from starting the
network again during this sessoin.

*** Starting CLI:
mininet> 
```

After the topology was created, the Floodlight Openflow SDN controller was implemented and chosen. Figure 2 shows the hosts connecting to the switches IPs using the TCP 3-way handshake process

Figure 1: Network Topology

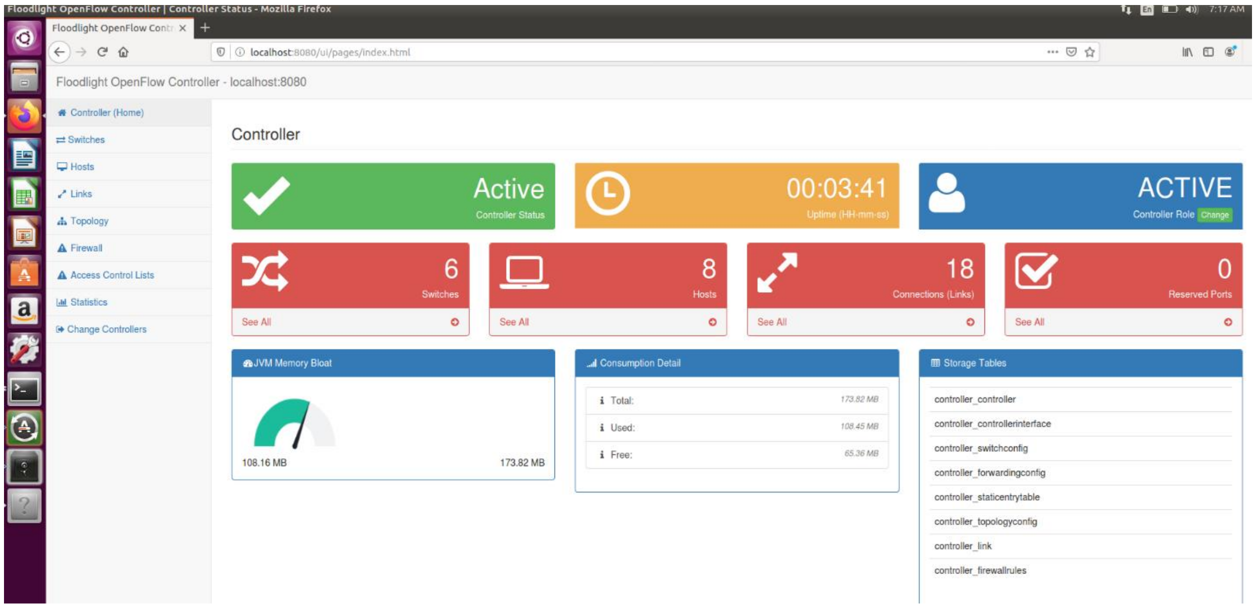


```
ubuntu@ubuntu1604: ~/Desktop/floodlight
2022-03-21 07:14:14.334 INFO  [n.f.f.Forwarding] Default flags will be set to SEND_FLOW_REM false.
2022-03-21 07:14:14.334 INFO  [n.f.f.Forwarding] Default flow matches set to: IN_PORT=true, VLAN=true, MAC=true, IP=true, FLAG=true, TPPT=true
2022-03-21 07:14:14.335 INFO  [n.f.f.Forwarding] Default detailed flow matches set to: SRC_MAC=true, DST_MAC=true, SRC_IP=true, DST_IP=true, SRC_TPPT=true, DST_TPPT=true
2022-03-21 07:14:14.335 INFO  [n.f.f.Forwarding] Not flooding ARP packets. ARP flows will be inserted for known destinations
2022-03-21 07:14:14.335 INFO  [n.f.f.Forwarding] Flows will be removed on link/port down events
2022-03-21 07:14:14.336 INFO  [n.f.s.StatisticsCollector] Statistics collection disabled
2022-03-21 07:14:14.336 INFO  [n.f.s.StatisticsCollector] Port statistics collection interval set to 10s
2022-03-21 07:14:14.598 INFO  [o.s.s.i.SyncManager] [1] Updating sync configuration ClusterConfig [allNodes={1=Node [hostname=192.168.1.100, port=6642, nodeId=1, domainId=1], 2=Node [hostname=192.168.1.10
0, port=6643, nodeId=2, domainId=1]}, authScheme=CHALLENGE_RESPONSE, keyStorePath=/etc/floodlight/key2.jceks, keyStorePassword is set]
2022-03-21 07:14:15.55 INFO  [o.s.s.i.r.RPCService] Listening for internal floodlight RPC on 0.0.0.0/0.0.0.0:6642
2022-03-21 07:14:15.909 INFO  [o.r.C.I.Server] Starting the Simple [HTTP/1.1] server on port 8080
2022-03-21 07:14:15.918 INFO  [org.restlet] Starting net.floodlightcontroller.restserver.RestApiServer$RestApplication application
2022-03-21 07:14:16.238 INFO  [n.f.c.i.OFChannelHandler] New switch connection from /127.0.0.1:45280
2022-03-21 07:14:16.239 INFO  [n.f.c.i.OFChannelHandler] New switch connection from /127.0.0.1:45286
2022-03-21 07:14:16.239 INFO  [n.f.c.i.OFChannelHandler] New switch connection from /127.0.0.1:45276
2022-03-21 07:14:16.238 INFO  [n.f.c.i.OFChannelHandler] New switch connection from /127.0.0.1:45282
2022-03-21 07:14:16.238 INFO  [n.f.c.i.OFChannelHandler] New switch connection from /127.0.0.1:45284
2022-03-21 07:14:16.238 INFO  [n.f.c.i.OFChannelHandler] New switch connection from /127.0.0.1:45278
2022-03-21 07:14:16.445 INFO  [n.f.c.i.OFChannelHandler] Negotiated OpenFlow version of OF_13 for /127.0.0.1:45280 with greatest version bitmap algorithm.
2022-03-21 07:14:16.446 INFO  [n.f.c.i.OFChannelHandler] Negotiated OpenFlow version of OF_13 for /127.0.0.1:45278 with greatest version bitmap algorithm.
2022-03-21 07:14:16.446 INFO  [n.f.c.i.OFChannelHandler] Negotiated OpenFlow version of OF_13 for /127.0.0.1:45276 with greatest version bitmap algorithm.
2022-03-21 07:14:16.456 INFO  [n.f.c.i.OFChannelHandler] Negotiated OpenFlow version of OF_13 for /127.0.0.1:45286 with greatest version bitmap algorithm.
2022-03-21 07:14:16.459 INFO  [n.f.c.i.OFChannelHandler] Negotiated OpenFlow version of OF_13 for /127.0.0.1:45284 with greatest version bitmap algorithm.
2022-03-21 07:14:16.459 INFO  [n.f.c.i.OFChannelHandler] Negotiated OpenFlow version of OF_13 for /127.0.0.1:45282 with greatest version bitmap algorithm.
2022-03-21 07:14:16.958 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Switch OFSwitch DPID[00:00:00:00:00:00:00:03] bound to class class net.floodlightcontroller.core.internal.OFSwitch, description SwitchDescr
iption [manufacturerDescription=Nicira, Inc., hardwareDescription=Open vSwitch, softwareDescription=2.5.5, serialNumber=None, datapathDescription=None]
2022-03-21 07:14:16.959 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Switch OFSwitch DPID[00:00:00:00:00:00:00:04] bound to class class net.floodlightcontroller.core.internal.OFSwitch, description SwitchDescr
iption [manufacturerDescription=Nicira, Inc., hardwareDescription=Open vSwitch, softwareDescription=2.5.5, serialNumber=None, datapathDescription=None]
2022-03-21 07:14:16.959 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Switch OFSwitch DPID[00:00:00:00:00:00:00:06] bound to class class net.floodlightcontroller.core.internal.OFSwitch, description SwitchDescr
iption [manufacturerDescription=Nicira, Inc., hardwareDescription=Open vSwitch, softwareDescription=2.5.5, serialNumber=None, datapathDescription=None]
2022-03-21 07:14:16.961 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Switch OFSwitch DPID[00:00:00:00:00:00:00:01] bound to class class net.floodlightcontroller.core.internal.OFSwitch, description SwitchDescr
iption [manufacturerDescription=Nicira, Inc., hardwareDescription=Open vSwitch, softwareDescription=2.5.5, serialNumber=None, datapathDescription=None]
2022-03-21 07:14:16.962 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Switch OFSwitch DPID[00:00:00:00:00:00:00:02] bound to class class net.floodlightcontroller.core.internal.OFSwitch, description SwitchDescr
iption [manufacturerDescription=Nicira, Inc., hardwareDescription=Open vSwitch, softwareDescription=2.5.5, serialNumber=None, datapathDescription=None]
2022-03-21 07:14:17.18 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Switch OFSwitch DPID[00:00:00:00:00:00:00:05] bound to class class net.floodlightcontroller.core.internal.OFSwitch, description SwitchDescri
ption [manufacturerDescription=Nicira, Inc., hardwareDescription=Open vSwitch, softwareDescription=2.5.5, serialNumber=None, datapathDescription=None]
2022-03-21 07:14:19.477 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Defining switch role from config file: ROLE_MASTER
2022-03-21 07:14:19.478 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Defining switch role from config file: ROLE_MASTER
2022-03-21 07:14:19.479 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Defining switch role from config file: ROLE_MASTER
2022-03-21 07:14:19.480 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Defining switch role from config file: ROLE_MASTER
2022-03-21 07:14:19.520 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Clearing flow tables of 00:00:00:00:00:00:00:03 on upcoming transition to MASTER.
2022-03-21 07:14:19.521 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Clearing flow tables of 00:00:00:00:00:00:00:01 on upcoming transition to MASTER.
2022-03-21 07:14:19.521 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Clearing flow tables of 00:00:00:00:00:00:00:04 on upcoming transition to MASTER.
2022-03-21 07:14:19.521 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Clearing flow tables of 00:00:00:00:00:00:00:05 on upcoming transition to MASTER.
2022-03-21 07:14:19.558 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Defining switch role from config file: ROLE_MASTER
2022-03-21 07:14:19.559 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Clearing flow tables of 00:00:00:00:00:00:00:02 on upcoming transition to MASTER.
2022-03-21 07:14:19.571 INFO  [n.f.c.i.OFSwitchHandshakeHandler] Clearing flow tables of 00:00:00:00:00:00:00:06 on upcoming transition to MASTER.
2022-03-21 07:14:19.898 ERROR [n.f.c.i.OFSwitchHandshakeHandler] OFGroupModFailedErrorMsgVer13(xid=76, code=INVALID_GROUP, data=OFGroupDeleteVer13(xid=76, groupType=INDIRECT, group=all, buckets=[])) from
switch OFSwitch DPID[00:00:00:00:00:00:00:01] in state net.floodlightcontroller.core.internal.OFSwitchHandshakeHandler$MasterState@28c53e50
```

Figure 2: Topology Connections

Figure 3: Floodlight GUI

In contrast, another popular controller is Opendaylight, which is similar to Floodlight and can provide northbound services to multiple stacks and of neutron networks. However, Floodlight provides more efficient services in terms of packet loss and latency.



Floodlight OpenFlow Controller | Controller Status - Mozilla Firefox

localhost:8080/ui/pages/index.html

Floodlight OpenFlow Controller - localhost:8080

Controller

Active
Controller Status

00:03:41
Uptime (HH-mm-ss)

ACTIVE
Controller Role

6
Switches

8
Hosts

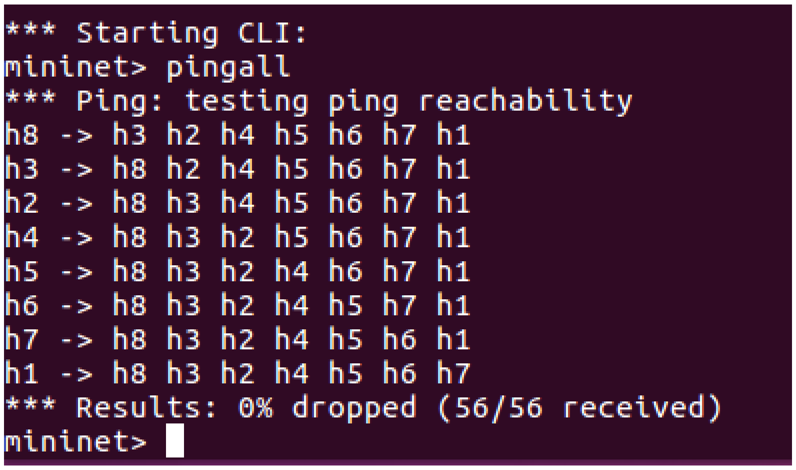
18
Connections (Links)

0
Reserved Ports

JVM Memory Bloat
108.16 MB
173.82 MB

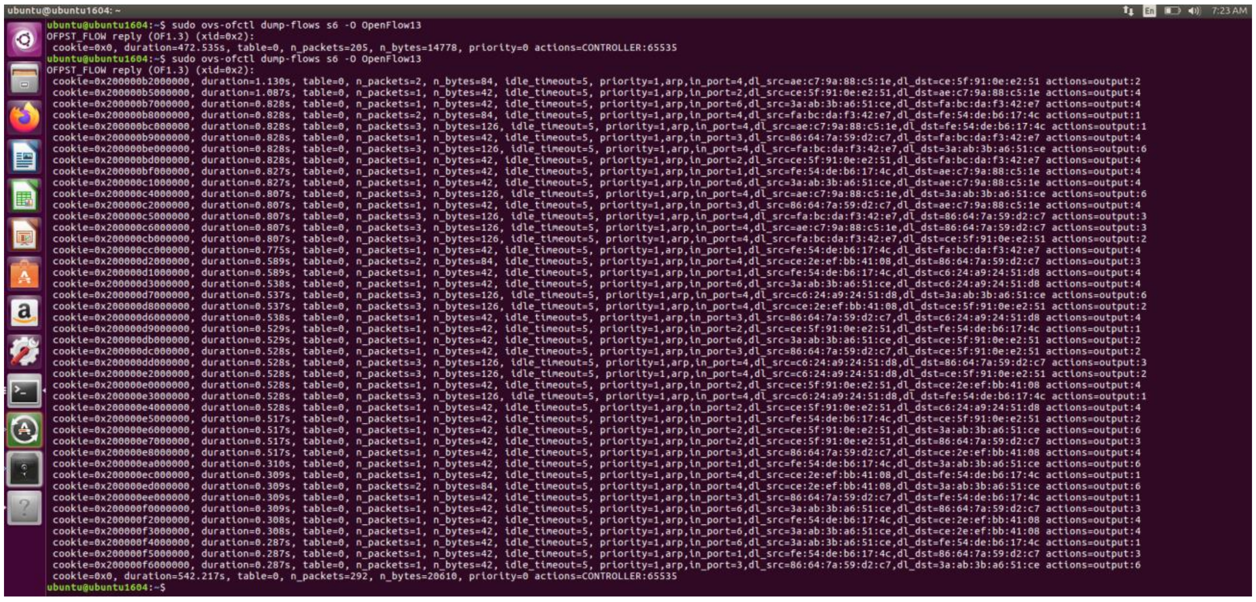
Consumption Detail
Total: 173.82 MB
Used: 108.45 MB
Free: 65.36 MB

Storage Tables
controller_controller
controller_controllerinterface
controller_switchconfig
controller_forwardingconfig
controller_staticentrytable
controller_topologyconfig
controller_link
controller_firewallrules



```
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h8 -> h3 h2 h4 h5 h6 h7 h1
h3 -> h8 h2 h4 h5 h6 h7 h1
h2 -> h8 h3 h4 h5 h6 h7 h1
h4 -> h8 h3 h2 h5 h6 h7 h1
h5 -> h8 h3 h2 h4 h6 h7 h1
h6 -> h8 h3 h2 h4 h5 h7 h1
h7 -> h8 h3 h2 h4 h5 h6 h1
h1 -> h8 h3 h2 h4 h5 h6 h7
*** Results: 0% dropped (56/56 received)
mininet>
```

Figure 4: Topology PingAll Test

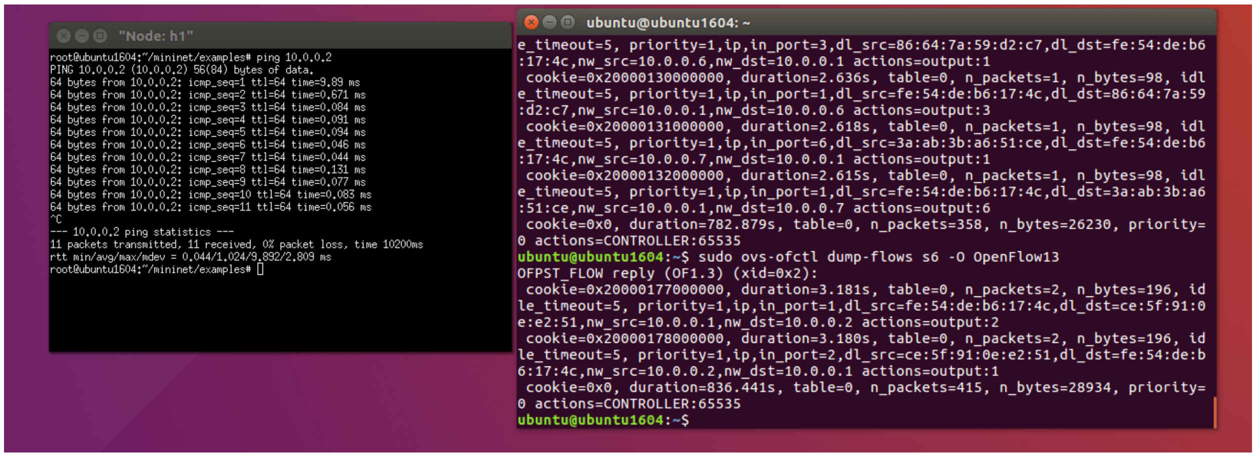


```
ubuntu@ubuntu1604: ~
ubuntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=472.535s, table=0, n_packets=205, n_bytes=14778, priority=0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x200000b2000000, duration=1.130s, table=0, n_packets=2, n_bytes=84, idle_timeout=5, priority=1,arp,in_port=4,dl_src=ae:c7:9a:88:c5:1e,dl_dst=ce:5f:91:0e:e2:51 actions=output:2
 cookie=0x200000b5000000, duration=1.087s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=ce:5f:91:0e:e2:51,dl_dst=ae:c7:9a:88:c5:1e actions=output:4
 cookie=0x200000b7000000, duration=0.828s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=6,dl_src=3a:ab:3b:a6:51:ce,dl_dst=fa:bc:da:f3:42:e7 actions=output:4
 cookie=0x200000b8000000, duration=0.828s, table=0, n_packets=2, n_bytes=84, idle_timeout=5, priority=1,arp,in_port=4,dl_src=fa:bc:da:f3:42:e7,dl_dst=fe:54:de:b6:17:4c actions=output:1
 cookie=0x200000bc000000, duration=0.828s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=ae:c7:9a:88:c5:1e,dl_dst=fe:54:de:b6:17:4c actions=output:1
 cookie=0x200000b9000000, duration=0.828s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=3,dl_src=86:64:7a:59:d2:c7,dl_dst=fa:bc:da:f3:42:e7 actions=output:4
 cookie=0x200000be000000, duration=0.828s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=fa:bc:da:f3:42:e7,dl_dst=3a:ab:3b:a6:51:ce actions=output:6
 cookie=0x200000bd000000, duration=0.828s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=ce:5f:91:0e:e2:51,dl_dst=fa:bc:da:f3:42:e7 actions=output:4
 cookie=0x200000bf000000, duration=0.827s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=ae:c7:9a:88:c5:1e actions=output:4
 cookie=0x200000c1000000, duration=0.827s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=6,dl_src=3a:ab:3b:a6:51:ce,dl_dst=ae:c7:9a:88:c5:1e actions=output:4
 cookie=0x200000c4000000, duration=0.807s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=ae:c7:9a:88:c5:1e,dl_dst=3a:ab:3b:a6:51:ce actions=output:6
 cookie=0x200000c2000000, duration=0.807s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=3,dl_src=86:64:7a:59:d2:c7,dl_dst=ae:c7:9a:88:c5:1e actions=output:4
 cookie=0x200000c5000000, duration=0.807s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=fa:bc:da:f3:42:e7,dl_dst=86:64:7a:59:d2:c7 actions=output:3
 cookie=0x200000c6000000, duration=0.807s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=ae:c7:9a:88:c5:1e,dl_dst=86:64:7a:59:d2:c7 actions=output:3
 cookie=0x200000cb000000, duration=0.807s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=fa:bc:da:f3:42:e7,dl_dst=ce:5f:91:0e:e2:51 actions=output:2
 cookie=0x200000cc000000, duration=0.775s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=fa:bc:da:f3:42:e7 actions=output:4
 cookie=0x200000d2000000, duration=0.589s, table=0, n_packets=2, n_bytes=84, idle_timeout=5, priority=1,arp,in_port=4,dl_src=ce:2e:ef:bb:41:08,dl_dst=86:64:7a:59:d2:c7 actions=output:3
 cookie=0x200000d1000000, duration=0.589s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=c6:24:a9:24:51:d8 actions=output:4
 cookie=0x200000d3000000, duration=0.538s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=6,dl_src=3a:ab:3b:a6:51:ce,dl_dst=c6:24:a9:24:51:d8 actions=output:4
 cookie=0x200000d7000000, duration=0.537s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=c6:24:a9:24:51:d8,dl_dst=3a:ab:3b:a6:51:ce actions=output:6
 cookie=0x200000d8000000, duration=0.537s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=ce:2e:ef:bb:41:08,dl_dst=ce:5f:91:0e:e2:51 actions=output:2
 cookie=0x200000d6000000, duration=0.538s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=3,dl_src=86:64:7a:59:d2:c7,dl_dst=c6:24:a9:24:51:d8 actions=output:4
 cookie=0x200000d9000000, duration=0.529s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=ce:5f:91:0e:e2:51,dl_dst=fe:54:de:b6:17:4c actions=output:1
 cookie=0x200000db000000, duration=0.529s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=6,dl_src=3a:ab:3b:a6:51:ce,dl_dst=ce:5f:91:0e:e2:51 actions=output:2
 cookie=0x200000dc000000, duration=0.528s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=3,dl_src=86:64:7a:59:d2:c7,dl_dst=ce:5f:91:0e:e2:51 actions=output:2
 cookie=0x200000dd000000, duration=0.528s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=c6:24:a9:24:51:d8,dl_dst=86:64:7a:59:d2:c7 actions=output:3
 cookie=0x200000e2000000, duration=0.528s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=c6:24:a9:24:51:d8,dl_dst=ce:5f:91:0e:e2:51 actions=output:2
 cookie=0x200000e0000000, duration=0.528s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=ce:5f:91:0e:e2:51,dl_dst=ce:2e:ef:bb:41:08 actions=output:4
 cookie=0x200000e3000000, duration=0.528s, table=0, n_packets=3, n_bytes=126, idle_timeout=5, priority=1,arp,in_port=4,dl_src=c6:24:a9:24:51:d8,dl_dst=fe:54:de:b6:17:4c actions=output:1
 cookie=0x200000e4000000, duration=0.528s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=ce:5f:91:0e:e2:51,dl_dst=c6:24:a9:24:51:d8 actions=output:4
 cookie=0x200000e5000000, duration=0.517s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=ce:5f:91:0e:e2:51 actions=output:2
 cookie=0x200000e6000000, duration=0.517s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=ce:5f:91:0e:e2:51,dl_dst=3a:ab:3b:a6:51:ce actions=output:6
 cookie=0x200000e7000000, duration=0.517s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=ce:5f:91:0e:e2:51,dl_dst=86:64:7a:59:d2:c7 actions=output:3
 cookie=0x200000e8000000, duration=0.517s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=3,dl_src=86:64:7a:59:d2:c7,dl_dst=ce:2e:ef:bb:41:08 actions=output:4
 cookie=0x200000ea000000, duration=0.310s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=3a:ab:3b:a6:51:ce actions=output:6
 cookie=0x200000ec000000, duration=0.309s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=4,dl_src=ce:2e:ef:bb:41:08,dl_dst=fe:54:de:b6:17:4c actions=output:1
 cookie=0x200000ed000000, duration=0.309s, table=0, n_packets=2, n_bytes=84, idle_timeout=5, priority=1,arp,in_port=4,dl_src=ce:2e:ef:bb:41:08,dl_dst=3a:ab:3b:a6:51:ce actions=output:6
 cookie=0x200000ee000000, duration=0.309s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=3,dl_src=86:64:7a:59:d2:c7,dl_dst=fe:54:de:b6:17:4c actions=output:1
 cookie=0x200000f0000000, duration=0.309s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=6,dl_src=3a:ab:3b:a6:51:ce,dl_dst=86:64:7a:59:d2:c7 actions=output:3
 cookie=0x200000f2000000, duration=0.308s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=ce:2e:ef:bb:41:08 actions=output:4
 cookie=0x200000f3000000, duration=0.308s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=6,dl_src=3a:ab:3b:a6:51:ce,dl_dst=ce:2e:ef:bb:41:08 actions=output:4
 cookie=0x200000f4000000, duration=0.287s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=6,dl_src=3a:ab:3b:a6:51:ce,dl_dst=fe:54:de:b6:17:4c actions=output:1
 cookie=0x200000f5000000, duration=0.287s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=86:64:7a:59:d2:c7 actions=output:3
 cookie=0x200000f6000000, duration=0.287s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=3,dl_src=86:64:7a:59:d2:c7,dl_dst=3a:ab:3b:a6:51:ce actions=output:6
 cookie=0x0, duration=542.217s, table=0, n_packets=292, n_bytes=20610, priority=0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$
```

Figure 5: Topology Flow Table

Figure 3 shows the Floodlight GUI which has a visual representation of all the hosts and switches, and to test the network works, a ping all test was done in Mininet which resulted in 0 packet loss, showing the topology working correctly (Figure 4).

Figure 5 shows the topology flow table, meaning any packet information is displayed here and this further shows the network working correctly without any errors or drops.


"Node: h1"

```
root@ubuntu1604:~/mininet/examples# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=9.89 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.671 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.094 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.131 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.077 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.056 ms
^C
--- 10.0.0.2 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10200ms
rtt min/avg/max/mdev = 0.044/1.024/9.892/2.809 ms
root@ubuntu1604:~/mininet/examples# 
```

ubuntu@ubuntu1604: ~

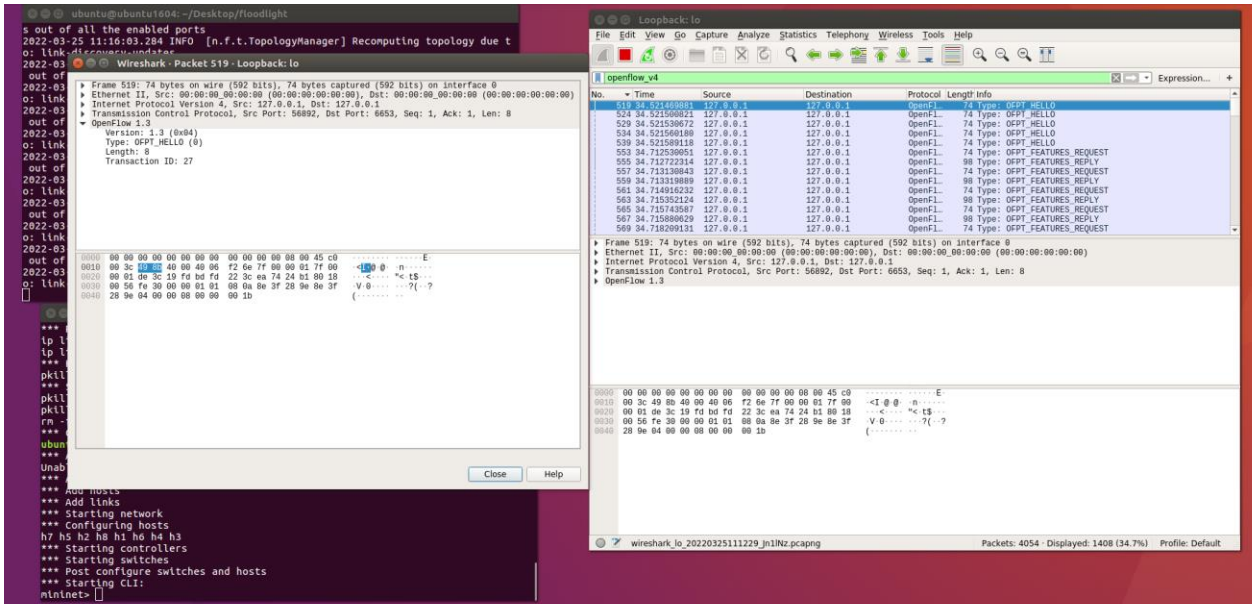
```
e_timeout=5, priority=1,ip,in_port=3,dl_src=86:64:7a:59:d2:c7,dl_dst=fe:54:de:b6
:17:4c,nw_src=10.0.0.6,nw_dst=10.0.0.1 actions=output:1
 cookie=0x20000130000000, duration=2.636s, table=0, n_packets=1, n_bytes=98, idl
e_timeout=5, priority=1,ip,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=86:64:7a:59
:d2:c7,nw_src=10.0.0.1,nw_dst=10.0.0.6 actions=output:3
 cookie=0x20000131000000, duration=2.618s, table=0, n_packets=1, n_bytes=98, idl
e_timeout=5, priority=1,ip,in_port=6,dl_src=3a:ab:3b:a6:51:ce,dl_dst=fe:54:de:b6
:17:4c,nw_src=10.0.0.7,nw_dst=10.0.0.1 actions=output:1
 cookie=0x20000132000000, duration=2.615s, table=0, n_packets=1, n_bytes=98, idl
e_timeout=5, priority=1,ip,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=3a:ab:3b:a6
:51:ce,nw_src=10.0.0.1,nw_dst=10.0.0.7 actions=output:6
 cookie=0x0, duration=782.879s, table=0, n_packets=358, n_bytes=26230, priority=
0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x20000177000000, duration=3.181s, table=0, n_packets=2, n_bytes=196, id
le_timeout=5, priority=1,ip,in_port=1,dl_src=fe:54:de:b6:17:4c,dl_dst=ce:5f:91:0
e:e2:51,nw_src=10.0.0.1,nw_dst=10.0.0.2 actions=output:2
 cookie=0x20000178000000, duration=3.180s, table=0, n_packets=2, n_bytes=196, id
le_timeout=5, priority=1,ip,in_port=2,dl_src=ce:5f:91:0e:e2:51,dl_dst=fe:54:de:b
6:17:4c,nw_src=10.0.0.2,nw_dst=10.0.0.1 actions=output:1
 cookie=0x0, duration=836.441s, table=0, n_packets=415, n_bytes=28934, priority=
0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$
```

Figure 6: OpenFlow Table Test

Figure 6 represents the test done to achieve the flow tables, and that each individual node can ping each other, concluding the network is setup accurately with no packet loss.

2.2. OpenFlow Packet Analysis

The Openflow packet analysis was achieved using Wireshark, where the Floodlight controller was started, and traffic was filtered to Loopback via openflow_v4. This is because the network uses the Openflow framework and therefore this filter achieves relevant traffic (figure 7).



ubuntu@ubuntu1604: ~/Desktop/floodlight
Wireshark · Packet 519 · Loopback: lo
Close
Help
Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

```
s out of all the enabled ports
2022-03-25 11:16:03.284 INFO  [n.f.t.TopologyManager] Recomputing topology due t
o: link-discovery-updates
2022-03
 out of
2022-03
o: link
2022-03
 out of
2022-03
o: link
2022-03
 out of
2022-03
o: link
2022-03
 out of
2022-03
o: link
2022-03
 out of
2022-03
o: link
```

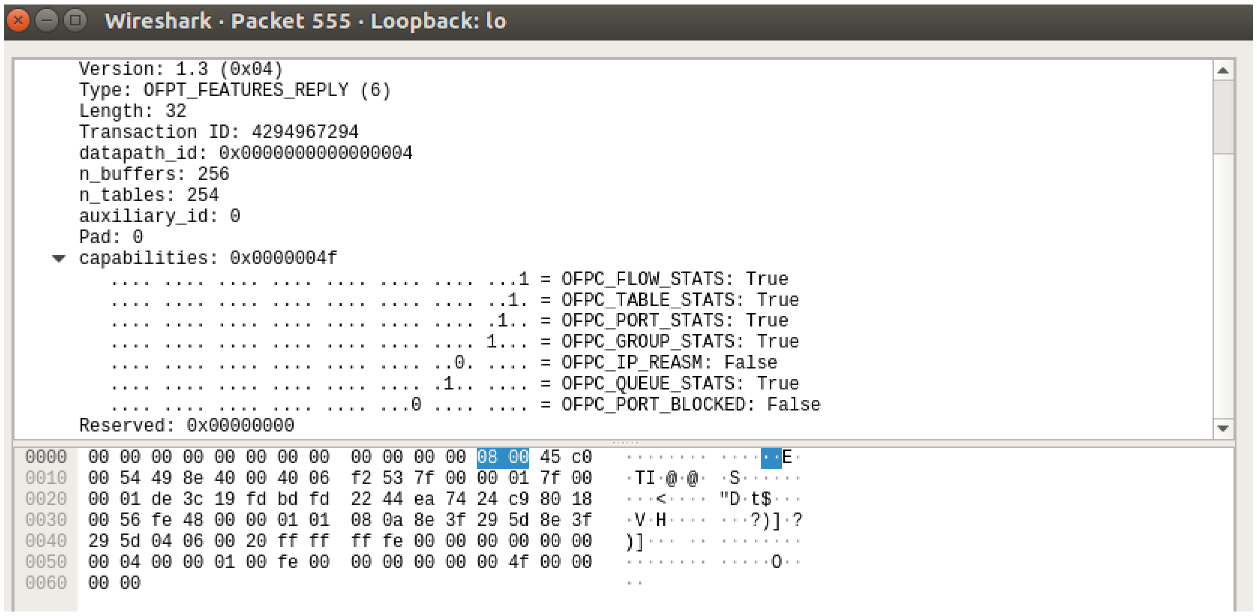
```
▸ Frame 519: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 56892, Dst Port: 6653, Seq: 1, Ack: 1, Len: 8
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_HELLO (0)
    Length: 8
    Transaction ID: 27

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ········ ······E·
0010  00 3c 49 8b 40 00 40 06  f2 6e 7f 00 00 01 7f 00   ·<I·@·@· ·n······
0020  00 01 de 3c 19 fd bd fd  22 3c ea 74 24 b1 80 18   ···<···· "<·t$···
0030  00 56 fe 30 00 00 01 01  08 0a 8e 3f 28 9e 8e 3f   ·V·0···· ···?(··?
0040  28 9e 04 00 00 08 00 00  00 1b                     (······· ··
```

```
***
ip l
ip l
***
pkil
***
pkil
pkil
rm -
***
ubun
***
Unab
***
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h7 h5 h2 h8 h1 h6 h4 h3
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet>
```

```
openflow_v4

No.   Time          Source      Destination  Protocol  Length Info
  519 34.521469881  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  524 34.521500821  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  529 34.521530672  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  534 34.521560180  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  539 34.521589118  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  553 34.712530051  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
  555 34.712722314  127.0.0.1   127.0.0.1    OpenFl…   98 Type: OFPT_FEATURES_REPLY
  557 34.713130843  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
  559 34.713319889  127.0.0.1   127.0.0.1    OpenFl…   98 Type: OFPT_FEATURES_REPLY
  561 34.714916232  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
  563 34.715352124  127.0.0.1   127.0.0.1    OpenFl…   98 Type: OFPT_FEATURES_REPLY
  565 34.715743587  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
  567 34.715880629  127.0.0.1   127.0.0.1    OpenFl…   98 Type: OFPT_FEATURES_REPLY
  569 34.718209131  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST

▸ Frame 519: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 56892, Dst Port: 6653, Seq: 1, Ack: 1, Len: 8
▸ OpenFlow 1.3

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ········ ······E·
0010  00 3c 49 8b 40 00 40 06  f2 6e 7f 00 00 01 7f 00   ·<I·@·@· ·n······
0020  00 01 de 3c 19 fd bd fd  22 3c ea 74 24 b1 80 18   ···<···· "<·t$···
0030  00 56 fe 30 00 00 01 01  08 0a 8e 3f 28 9e 8e 3f   ·V·0···· ···?(··?
0040  28 9e 04 00 00 08 00 00  00 1b                     (······· ··

wireshark_lo_20220325111229_Jn1lNz.pcapng    Packets: 4054 · Displayed: 1408 (34.7%)    Profile: Default
```

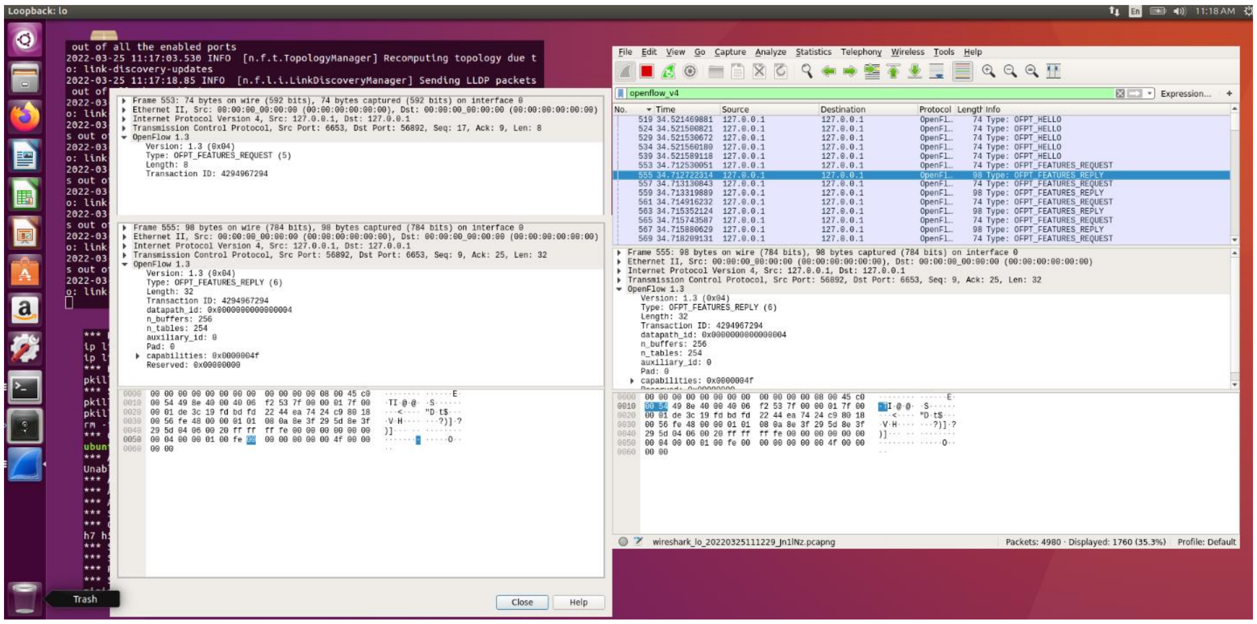
Figure 7: Wireshark OFPT_Hello

OFPT_Hello establishes the handshake when an Openflow connection is confirmed and contains elements to help the connection setup. The Floodlight controller and switch uses OFPT_Hello to also establish a familiar version number in order for messages to be communicated across the network.



```
      Version: 1.3 (0x04)
      Type: OFPT_FEATURES_REPLY (6)
      Length: 32
      Transaction ID: 4294967294
      datapath_id: 0x0000000000000004
      n_buffers: 256
      n_tables: 254
      auxiliary_id: 0
      Pad: 0
    ▾ capabilities: 0x0000004f
         .... .... .... .... .... .... .... ...1 = OFPC_FLOW_STATS: True
         .... .... .... .... .... .... .... ..1. = OFPC_TABLE_STATS: True
         .... .... .... .... .... .... .... .1.. = OFPC_PORT_STATS: True
         .... .... .... .... .... .... .... 1... = OFPC_GROUP_STATS: True
         .... .... .... .... .... .... ..0. .... = OFPC_IP_REASM: False
         .... .... .... .... .... .... .1.. .... = OFPC_QUEUE_STATS: True
         .... .... .... .... .... ...0 .... .... = OFPC_PORT_BLOCKED: False
      Reserved: 0x00000000
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ········ ······E·
0010  00 54 49 8e 40 00 40 06  f2 53 7f 00 00 01 7f 00   ·TI·@·@· ·S······
0020  00 01 de 3c 19 fd bd fd  22 44 ea 74 24 c9 80 18   ···<···· "D·t$···
0030  00 56 fe 48 00 00 01 01  08 0a 8e 3f 29 5d 8e 3f   ·V·H···· ···?)]·?
0040  29 5d 04 06 00 20 ff ff  ff fe 00 00 00 00 00 00   )]··· ·· ········
0050  00 04 00 00 01 00 fe 00  00 00 00 00 00 4f 00 00   ········ ·····O··
0060  00 00                                              ··
```

Figure 8: Wireshark OFPT_Features_Reply



```
 out of all the enabled ports
2022-03-25 11:17:03.530 INFO  [n.f.t.TopologyManager] Recomputing topology due t
o: link-discovery-updates
2022-03-25 11:17:18.85 INFO  [n.f.l.i.LinkDiscoveryManager] Sending LLDP packets
```

```
▸ Frame 553: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 6653, Dst Port: 56892, Seq: 17, Ack: 9, Len: 8
▾ OpenFlow 1.3
     Version: 1.3 (0x04)
     Type: OFPT_FEATURES_REQUEST (5)
     Length: 8
     Transaction ID: 4294967294
```

```
▸ Frame 555: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 56892, Dst Port: 6653, Seq: 9, Ack: 25, Len: 32
▾ OpenFlow 1.3
     Version: 1.3 (0x04)
     Type: OFPT_FEATURES_REPLY (6)
     Length: 32
     Transaction ID: 4294967294
     datapath_id: 0x0000000000000004
     n_buffers: 256
     n_tables: 254
     auxiliary_id: 0
     Pad: 0
   ▸ capabilities: 0x0000004f
     Reserved: 0x00000000
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ········ ······E·
0010  00 54 49 8e 40 00 40 06  f2 53 7f 00 00 01 7f 00   ·TI·@·@· ·S······
0020  00 01 de 3c 19 fd bd fd  22 44 ea 74 24 c9 80 18   ···<···· "D·t$···
0030  00 56 fe 48 00 00 01 01  08 0a 8e 3f 29 5d 8e 3f   ·V·H···· ···?)]·?
0040  29 5d 04 06 00 20 ff ff  ff fe 00 00 00 00 00 00   )]··· ·· ········
0050  00 04 00 00 01 00 fe 00  00 00 00 00 00 4f 00 00   ········ ·····O··
0060  00 00                                              ··
```

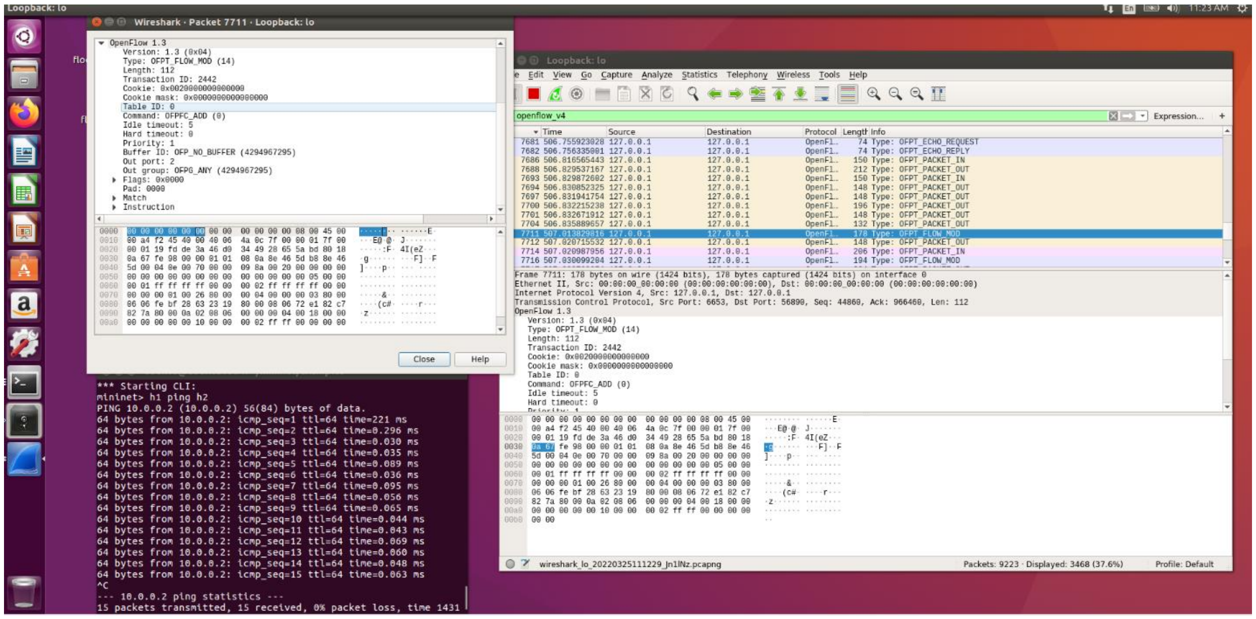
Close
Help
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
11:18 AM

```
openflow_v4
No.   Time          Source      Destination  Protocol  Length Info
  519 34.521469881  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  524 34.521500821  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  529 34.521530672  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  534 34.521560180  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  539 34.521589118  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_HELLO
  553 34.712530051  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
  555 34.712722314  127.0.0.1   127.0.0.1    OpenFl…   98 Type: OFPT_FEATURES_REPLY
  557 34.713130843  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
  559 34.713319889  127.0.0.1   127.0.0.1    OpenFl…   98 Type: OFPT_FEATURES_REPLY
  561 34.714916232  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
  563 34.715352124  127.0.0.1   127.0.0.1    OpenFl…   98 Type: OFPT_FEATURES_REPLY
  565 34.715743587  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
  567 34.715880629  127.0.0.1   127.0.0.1    OpenFl…   98 Type: OFPT_FEATURES_REPLY
  569 34.718209131  127.0.0.1   127.0.0.1    OpenFl…   74 Type: OFPT_FEATURES_REQUEST
```

```
▸ Frame 555: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 56892, Dst Port: 6653, Seq: 9, Ack: 25, Len: 32
▾ OpenFlow 1.3
     Version: 1.3 (0x04)
     Type: OFPT_FEATURES_REPLY (6)
     Length: 32
     Transaction ID: 4294967294
     datapath_id: 0x0000000000000004
     n_buffers: 256
     n_tables: 254
     auxiliary_id: 0
     Pad: 0
   ▸ capabilities: 0x0000004f
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ········ ······E·
0010  00 54 49 8e 40 00 40 06  f2 53 7f 00 00 01 7f 00   ·TI·@·@· ·S······
0020  00 01 de 3c 19 fd bd fd  22 44 ea 74 24 c9 80 18   ···<···· "D·t$···
0030  00 56 fe 48 00 00 01 01  08 0a 8e 3f 29 5d 8e 3f   ·V·H···· ···?)]·?
0040  29 5d 04 06 00 20 ff ff  ff fe 00 00 00 00 00 00   )]··· ·· ········
0050  00 04 00 00 01 00 fe 00  00 00 00 00 00 4f 00 00   ········ ·····O··
0060  00 00                                              ··

wireshark_lo_20220325111229_Jn1lNz.pcapng    Packets: 4980 · Displayed: 1760 (35.3%)    Profile: Default
```

Figure 9: Wireshark OFPT_Features_Request

Figures 8 and 9 exhibit the Openflow features messages, which are both used by the controller. The features request identifies the switches on a network and reads its basic attributes, consequently leading to the session being created simultaneously. The features reply establishes and confirms that the switch is connected to the controller and includes STP if the switch supports it.



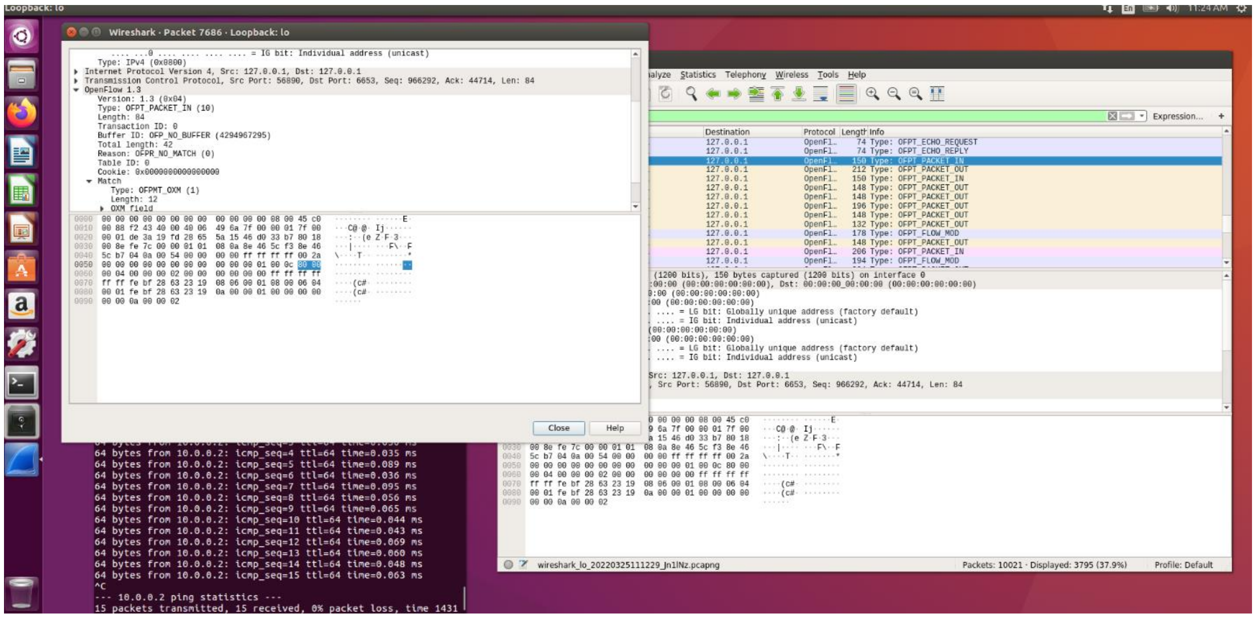
```
Wireshark · Packet 7711 · Loopback: lo

▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_FLOW_MOD (14)
    Length: 112
    Transaction ID: 2442
    Cookie: 0x0020000000000000
    Cookie mask: 0x0000000000000000
    Table ID: 0
    Command: OFPFC_ADD (0)
    Idle timeout: 5
    Hard timeout: 0
    Priority: 1
    Buffer ID: OFP_NO_BUFFER (4294967295)
    Out port: 2
    Out group: OFPG_ANY (4294967295)
  ▸ Flags: 0x0000
    Pad: 0000
  ▸ Match
  ▸ Instruction

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
0010  00 a4 f2 45 40 00 40 06  4a 0c 7f 00 00 01 7f 00   ···E@·@· J·······
0020  00 01 19 fd de 3a 46 d0  34 49 28 65 5a bd 80 18   ·····:F· 4I(eZ···
0030  0a 67 fe 98 00 00 01 01  08 0a 8e 46 5d b8 8e 46   ·g······ ···F]··F
0040  5d 00 04 0e 00 70 00 00  09 8a 00 20 00 00 00 00   ]····p·· ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 05 00 00   ········ ········
0060  00 01 ff ff ff ff 00 00  00 02 ff ff ff ff 00 00   ········ ········
0070  00 00 00 01 00 26 80 00  00 04 00 00 00 03 80 00   ·····&·· ········
0080  06 06 fe bf 28 63 23 19  80 00 08 06 72 e1 82 c7   ····(c#· ····r···
0090  82 7a 80 00 0a 02 08 06  00 00 00 04 00 18 00 00   ·z······ ········
00a0  00 00 00 00 00 10 00 00  00 02 ff ff 00 00 00 00   ········ ········
```

```
Loopback: lo
openflow_v4

     Time           Source     Destination  Protocol  Length Info
7681 506.755923028 127.0.0.1  127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7682 506.756335001 127.0.0.1  127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REPLY
7686 506.816565443 127.0.0.1  127.0.0.1    OpenFl…    150 Type: OFPT_PACKET_IN
7688 506.829537167 127.0.0.1  127.0.0.1    OpenFl…    212 Type: OFPT_PACKET_OUT
7693 506.829872602 127.0.0.1  127.0.0.1    OpenFl…    150 Type: OFPT_PACKET_IN
7694 506.830852325 127.0.0.1  127.0.0.1    OpenFl…    148 Type: OFPT_PACKET_OUT
7697 506.831941754 127.0.0.1  127.0.0.1    OpenFl…    148 Type: OFPT_PACKET_OUT
7700 506.832215238 127.0.0.1  127.0.0.1    OpenFl…    196 Type: OFPT_PACKET_OUT
7701 506.832671912 127.0.0.1  127.0.0.1    OpenFl…    148 Type: OFPT_PACKET_OUT
7704 506.835889657 127.0.0.1  127.0.0.1    OpenFl…    132 Type: OFPT_PACKET_OUT
7711 507.013829816 127.0.0.1  127.0.0.1    OpenFl…    178 Type: OFPT_FLOW_MOD
7712 507.020715532 127.0.0.1  127.0.0.1    OpenFl…    148 Type: OFPT_PACKET_OUT
7714 507.020987956 127.0.0.1  127.0.0.1    OpenFl…    206 Type: OFPT_PACKET_IN
7716 507.030099204 127.0.0.1  127.0.0.1    OpenFl…    194 Type: OFPT_FLOW_MOD

Frame 7711: 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits) on interface 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 6653, Dst Port: 56890, Seq: 44860, Ack: 966460, Len: 112
OpenFlow 1.3
  Version: 1.3 (0x04)
  Type: OFPT_FLOW_MOD (14)
  Length: 112
  Transaction ID: 2442
  Cookie: 0x0020000000000000
  Cookie mask: 0x0000000000000000
  Table ID: 0
  Command: OFPFC_ADD (0)
  Idle timeout: 5
  Hard timeout: 0

wireshark_lo_20220325111229_Jn1lNz.pcapng     Packets: 9223 · Displayed: 3468 (37.6%)     Profile: Default
```

```
*** Starting CLI:
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=221 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.296 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.063 ms
^C
--- 10.0.0.2 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 1431
```

Figure 10: Wireshark OFPT_Flow_Mod

Flow table modifications and configuration entries can be adjusted through the FLOW_MOD traffic request. This is carried out from the controller, and therefore any new entries to the specified flow table are through this request (Figure 10).

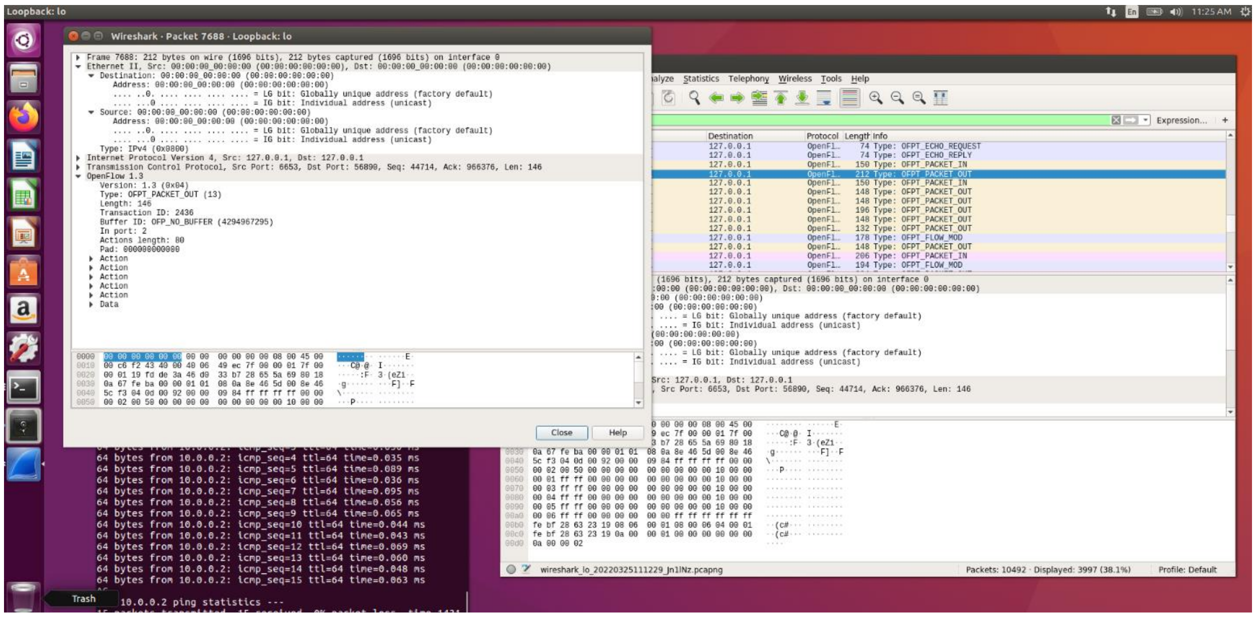


```
Wireshark · Packet 7686 · Loopback: lo

      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 56890, Dst Port: 6653, Seq: 966292, Ack: 44714, Len: 84
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_PACKET_IN (10)
    Length: 84
    Transaction ID: 0
    Buffer ID: OFP_NO_BUFFER (4294967295)
    Total length: 42
    Reason: OFPR_NO_MATCH (0)
    Table ID: 0
    Cookie: 0x0000000000000000
  ▾ Match
      Type: OFPMT_OXM (1)
      Length: 12
    ▸ OXM field

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 c0   ········ ······E·
0010  00 88 f2 43 40 00 40 06  49 6a 7f 00 00 01 7f 00   ···C@·@· Ij······
0020  00 01 de 3a 19 fd 28 65  5a 15 46 d0 33 b7 80 18   ···:··(e Z·F·3···
0030  00 8e fe 7c 00 00 01 01  08 0a 8e 46 5c f3 8e 46   ···|···· ···F\··F
0040  5c b7 04 0a 00 54 00 00  00 00 ff ff ff ff 00 2a   \····T·· ·······*
0050  00 00 00 00 00 00 00 00  00 00 00 01 00 0c 80 00   ········ ········
0060  00 04 00 00 00 02 00 00  00 00 00 00 ff ff ff ff   ········ ········
0070  ff ff fe bf 28 63 23 19  08 06 00 01 08 00 06 04   ····(c#· ········
0080  00 01 fe bf 28 63 23 19  0a 00 00 01 00 00 00 00   ····(c#· ········
0090  00 00 0a 00 00 02                                  ······
```

```
Destination  Protocol  Length Info
127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REPLY
127.0.0.1    OpenFl…    150 Type: OFPT_PACKET_IN
127.0.0.1    OpenFl…    212 Type: OFPT_PACKET_OUT
127.0.0.1    OpenFl…    150 Type: OFPT_PACKET_IN
127.0.0.1    OpenFl…    148 Type: OFPT_PACKET_OUT
127.0.0.1    OpenFl…    148 Type: OFPT_PACKET_OUT
127.0.0.1    OpenFl…    196 Type: OFPT_PACKET_OUT
127.0.0.1    OpenFl…    148 Type: OFPT_PACKET_OUT
127.0.0.1    OpenFl…    132 Type: OFPT_PACKET_OUT
127.0.0.1    OpenFl…    178 Type: OFPT_FLOW_MOD
127.0.0.1    OpenFl…    148 Type: OFPT_PACKET_OUT
127.0.0.1    OpenFl…    206 Type: OFPT_PACKET_IN
127.0.0.1    OpenFl…    194 Type: OFPT_FLOW_MOD

 (1200 bits), 150 bytes captured (1200 bits) on interface 0
:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
0:00 (00:00:00:00:00:00)
:00 (00:00:00:00:00:00)
. .... = LG bit: Globally unique address (factory default)
. .... = IG bit: Individual address (unicast)
(00:00:00:00:00:00)
:00 (00:00:00:00:00:00)
. .... = LG bit: Globally unique address (factory default)
. .... = IG bit: Individual address (unicast)

Src: 127.0.0.1, Dst: 127.0.0.1
, Src Port: 56890, Dst Port: 6653, Seq: 966292, Ack: 44714, Len: 84

wireshark_lo_20220325111229_Jn1lNz.pcapng     Packets: 10021 · Displayed: 3795 (37.9%)     Profile: Default
```

```
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.063 ms
^C
--- 10.0.0.2 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 1431
```

Figure 11: Wireshark OFPT_Packet_In



```
Wireshark · Packet 7688 · Loopback: lo

▸ Frame 7688: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface 0
▾ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
  ▾ Destination: 00:00:00_00:00:00 (00:00:00:00:00:00)
      Address: 00:00:00_00:00:00 (00:00:00:00:00:00)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: 00:00:00_00:00:00 (00:00:00:00:00:00)
      Address: 00:00:00_00:00:00 (00:00:00:00:00:00)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 6653, Dst Port: 56890, Seq: 44714, Ack: 966376, Len: 146
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_PACKET_OUT (13)
    Length: 146
    Transaction ID: 2436
    Buffer ID: OFP_NO_BUFFER (4294967295)
    In port: 2
    Actions length: 80
    Pad: 000000000000
  ▸ Action
  ▸ Action
  ▸ Action
  ▸ Action
  ▸ Action
  ▸ Data

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
0010  00 c6 f2 43 40 00 40 06  49 ec 7f 00 00 01 7f 00   ···C@·@· I·······
0020  00 01 19 fd de 3a 46 d0  33 b7 28 65 5a 69 80 18   ·····:F· 3·(eZi··
0030  0a 67 fe ba 00 00 01 01  08 0a 8e 46 5d 00 8e 46   ·g······ ···F]··F
0040  5c f3 04 0d 00 92 00 00  09 84 ff ff ff ff 00 00   \······· ········
0050  00 02 00 50 00 00 00 00  00 00 00 00 00 10 00 00   ···P···· ········
```

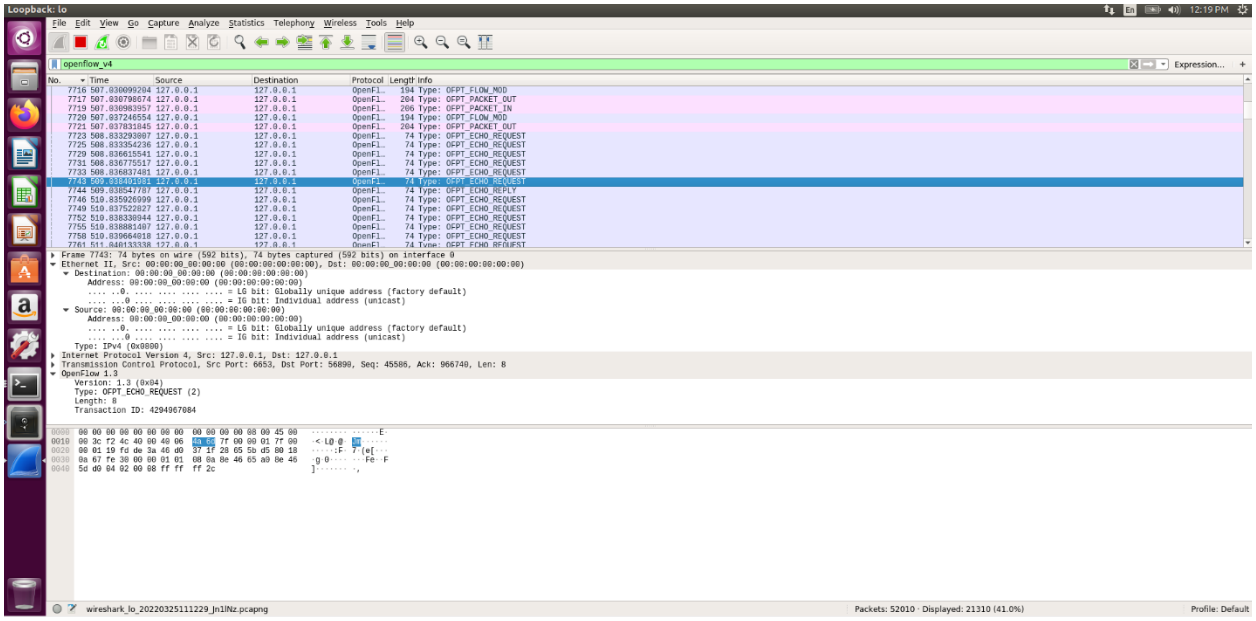
```
Destination   Protocol  Length Info
127.0.0.1     OpenFl…     74 Type: OFPT_ECHO_REQUEST
127.0.0.1     OpenFl…     74 Type: OFPT_ECHO_REPLY
127.0.0.1     OpenFl…    150 Type: OFPT_PACKET_IN
127.0.0.1     OpenFl…    212 Type: OFPT_PACKET_OUT
127.0.0.1     OpenFl…    150 Type: OFPT_PACKET_IN
127.0.0.1     OpenFl…    148 Type: OFPT_PACKET_OUT
127.0.0.1     OpenFl…    148 Type: OFPT_PACKET_OUT
127.0.0.1     OpenFl…    196 Type: OFPT_PACKET_OUT
127.0.0.1     OpenFl…    148 Type: OFPT_PACKET_OUT
127.0.0.1     OpenFl…    132 Type: OFPT_PACKET_OUT
127.0.0.1     OpenFl…    178 Type: OFPT_FLOW_MOD
127.0.0.1     OpenFl…    148 Type: OFPT_PACKET_OUT
127.0.0.1     OpenFl…    206 Type: OFPT_PACKET_IN
127.0.0.1     OpenFl…    194 Type: OFPT_FLOW_MOD
```

```
0030  0a 67 fe ba 00 00 01 01  08 0a 8e 46 5d 00 8e 46   ·g······ ···F]··F
0040  5c f3 04 0d 00 92 00 00  09 84 ff ff ff ff 00 00   \······· ········
0050  00 02 00 50 00 00 00 00  00 00 00 00 00 10 00 00   ···P···· ········
0060  00 01 ff ff 00 00 00 00  00 00 00 00 00 10 00 00   ········ ········
0070  00 03 ff ff 00 00 00 00  00 00 00 00 00 10 00 00   ········ ········
0080  00 04 ff ff 00 00 00 00  00 00 00 00 00 10 00 00   ········ ········
0090  00 05 ff ff 00 00 00 00  00 00 00 00 00 10 00 00   ········ ········
00a0  00 06 ff ff 00 00 00 00  00 00 ff ff ff ff ff ff   ········ ········
00b0  fe bf 28 63 23 19 08 06  00 01 08 00 06 04 00 01   ··(c#··· ········
00c0  fe bf 28 63 23 19 0a 00  00 01 00 00 00 00 00 00   ··(c#··· ········
00d0  0a 00 00 02                                        ····

wireshark_lo_20220325111229_Jn1lNz.pcapng     Packets: 10492 · Displayed: 3997 (38.1%)     Profile: Default
```

```
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.036 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.095 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.069 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.060 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.063 ms
^C
--- 10.0.0.2 ping statistics ---
```

Figure 12: Wireshark OFPT_Packet_Out

Figures 11 and 12 show the Openflow packet in and out messages. The ”packet in” messages use the data path to show packets received, which are then sent to the controller. The ”packet out” message is used when the controller sends a packet back through the data path.



```
openflow_v4

No.   Time        Source     Destination  Protocol  Length Info
7716 507.030099204 127.0.0.1 127.0.0.1    OpenFl…    194 Type: OFPT_FLOW_MOD
7717 507.030798674 127.0.0.1 127.0.0.1    OpenFl…    204 Type: OFPT_PACKET_OUT
7719 507.030983957 127.0.0.1 127.0.0.1    OpenFl…    206 Type: OFPT_PACKET_IN
7720 507.037246554 127.0.0.1 127.0.0.1    OpenFl…    194 Type: OFPT_FLOW_MOD
7721 507.037831845 127.0.0.1 127.0.0.1    OpenFl…    204 Type: OFPT_PACKET_OUT
7723 508.833293007 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7725 508.833354236 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7729 508.836615541 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7731 508.836775517 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7733 508.836837481 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7743 509.038401981 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7744 509.038547787 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REPLY
7746 510.835926999 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7749 510.837522827 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7752 510.838330944 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7755 510.838881407 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7758 510.839664018 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST
7761 511.040133338 127.0.0.1 127.0.0.1    OpenFl…     74 Type: OFPT_ECHO_REQUEST

▸ Frame 7743: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▾ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
  ▾ Destination: 00:00:00_00:00:00 (00:00:00:00:00:00)
      Address: 00:00:00_00:00:00 (00:00:00:00:00:00)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: 00:00:00_00:00:00 (00:00:00:00:00:00)
      Address: 00:00:00_00:00:00 (00:00:00:00:00:00)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 6653, Dst Port: 56890, Seq: 45586, Ack: 966740, Len: 8
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_ECHO_REQUEST (2)
    Length: 8
    Transaction ID: 4294967084

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
0010  00 3c f2 4c 40 00 40 06  4a 6d 7f 00 00 01 7f 00   ·<·L@·@· Jm······
0020  00 01 19 fd de 3a 46 d0  37 1f 28 65 5b d5 80 18   ·····:F· 7·(e[···
0030  0a 67 fe 30 00 00 01 01  08 0a 8e 46 65 a0 8e 46   ·g·0···· ···Fe··F
0040  5d d0 04 02 00 08 ff ff  ff 2c                     ]······· ·,

wireshark_lo_20220325111229_Jn1lNz.pcapng     Packets: 52010 · Displayed: 21310 (41.0%)     Profile: Default
```

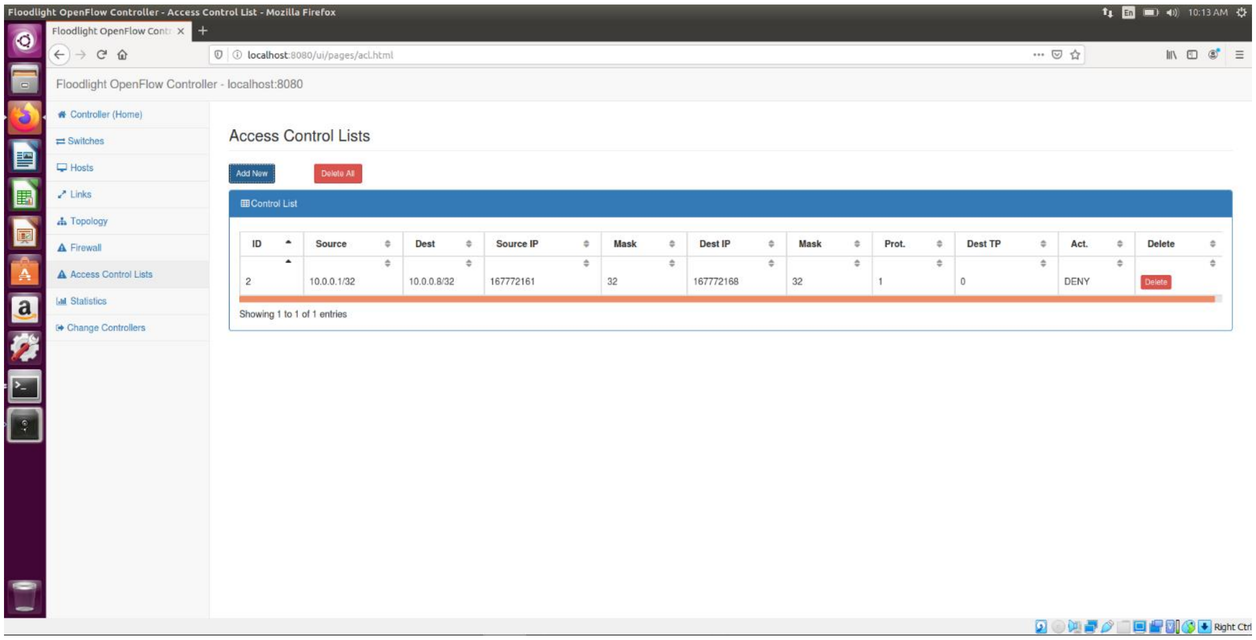
Figure 13: Wireshark OFPT_Echo_Request

Figure 13 shows the different types of echo-reply and sent messages across Wireshark. This means that the switch and controller can confirm an integrity between the connectivity of Openflow, to ensure connection is officially established. Each of these will help with connectivity troubleshooting and testing.

3. Network Functionalities

3.1. Firewall Implementation

The firewall security policy that will be applied will be done through the access control list (ACL) as shown in Figure 14. Within the network topology (figure 1), the hosts are split between different switches. These hosts could represent different departments and it’s possible to block specific traffic between these departments. Therefore, the ACL rule was created was to block ICMP traffic from the source, host 1, to the destination, host 8.


Floodlight OpenFlow Controller - Access Control List - Mozilla Firefox
localhost:8080/ui/pages/acl.html
Floodlight OpenFlow Controller - localhost:8080
Controller (Home)
Switches
Hosts
Links
Topology
Firewall
Access Control Lists
Statistics
Change Controllers
Add New
Delete All

Access Control Lists

Control List

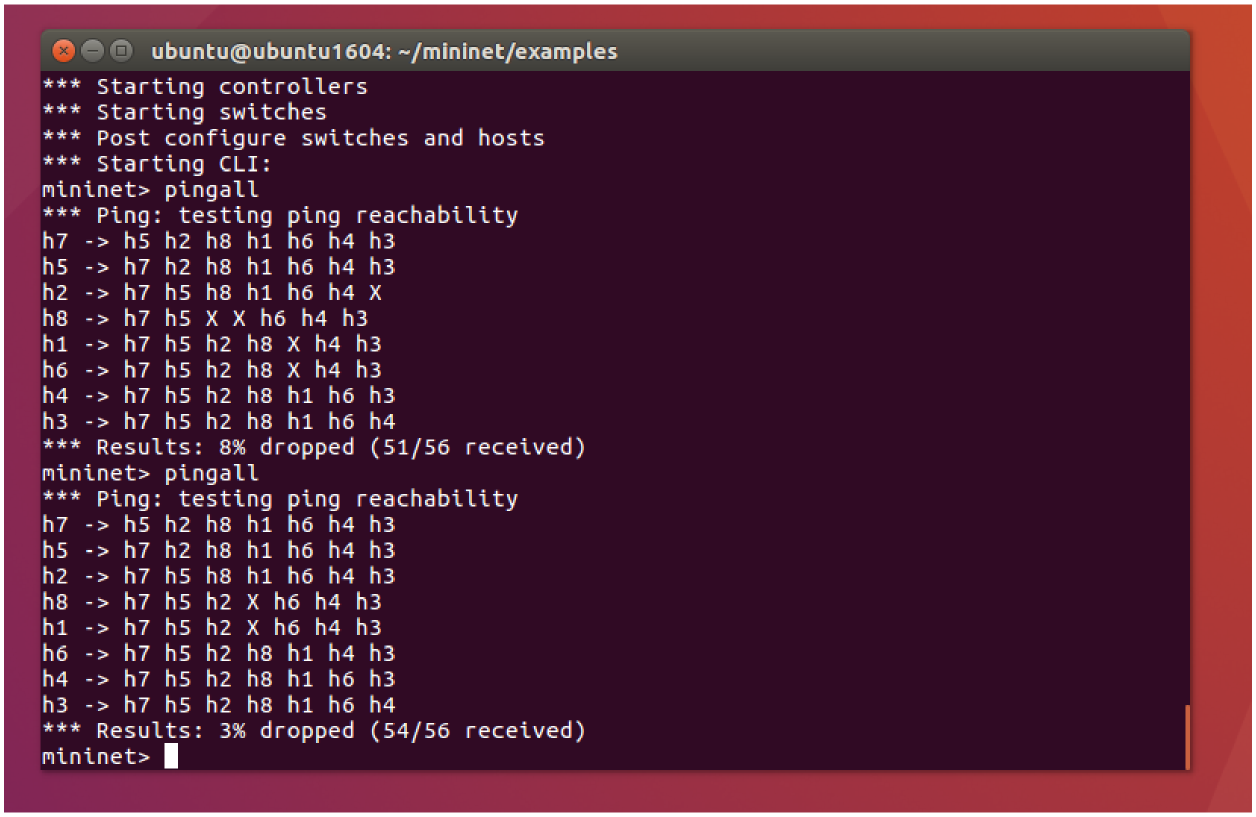
ID	Source	Dest	Source IP	Mask	Dest IP	Mask	Prot.	Dest TP	Act.	Delete
2	10.0.0.1/32	10.0.0.8/32	167772161	32	167772168	32	1	0	DENY	Delete

Showing 1 to 1 of 1 entries

Figure 14: ACL

This is significant because without ACL firewall policies, it would be very easy to one department to access other departments data, and that the ports within the network would be open to the whole company, creating a security risk based on the principle of least privilege. Additionally, if one host was to become hijacked, that potential attacker could access network traffic within the whole network.

Alternatively, a company could implement a firewall to improve security posture of the network. This is because firewalls perform stateful inspection, resulting in each packet being inspected per flow, whereas ACL are limited to stateless which is per packet. Furthermore, ACL do not include intrusion detection and generally only include layer 4, however, ACL is very easy to setup to create a blocked connection between 2 hosts.



```
ubuntu@ubuntu1604: ~/mininet/examples
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h7 -> h5 h2 h8 h1 h6 h4 h3
h5 -> h7 h2 h8 h1 h6 h4 h3
h2 -> h7 h5 h8 h1 h6 h4 X
h8 -> h7 h5 X X h6 h4 h3
h1 -> h7 h5 h2 h8 X h4 h3
h6 -> h7 h5 h2 h8 X h4 h3
h4 -> h7 h5 h2 h8 h1 h6 h3
h3 -> h7 h5 h2 h8 h1 h6 h4
*** Results: 8% dropped (51/56 received)
mininet> pingall
*** Ping: testing ping reachability
h7 -> h5 h2 h8 h1 h6 h4 h3
h5 -> h7 h2 h8 h1 h6 h4 h3
h2 -> h7 h5 h8 h1 h6 h4 h3
h8 -> h7 h5 h2 X h6 h4 h3
h1 -> h7 h5 h2 X h6 h4 h3
h6 -> h7 h5 h2 h8 h1 h4 h3
h4 -> h7 h5 h2 h8 h1 h6 h3
h3 -> h7 h5 h2 h8 h1 h6 h4
*** Results: 3% dropped (54/56 received)
mininet>
```

Figure 15: PingAll Test

Consequently, as a result of the ACL rule implementation between hosts 1 and 8, a Pingall test was carried out, as shown in Figure 15. This is to ensure that the ACL filter rule works correctly within the network. An incorrect test would mean that the ACL rule is working incorrectly, and therefore could be a security issue.

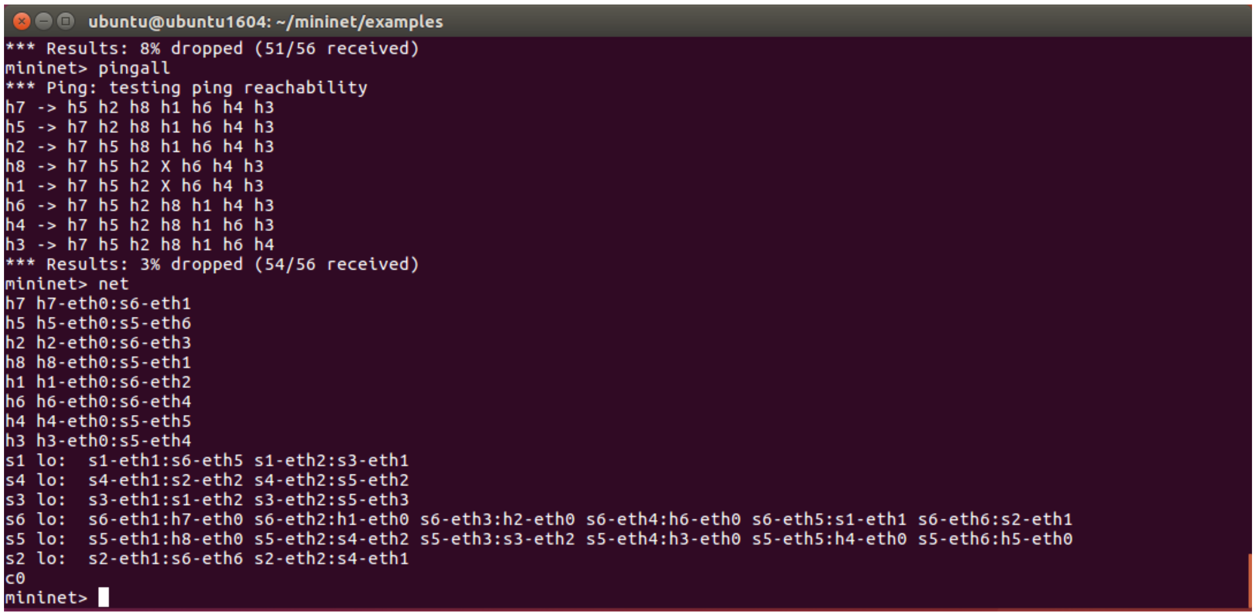
Using the command line within Ubuntu, the Pingall test within the Mininet interface resulted in only 2 packets dropped (figure 15), and those packets dropped are represented by the 'X', shown on host 1 to host 8, and on host 8 to host one, resulting in a successful ACL implementation.

3.2. Load Balancing

In order to balance network loads across distributed paths, load balancing needs to be introduced to the network. A company with 250 employees simultaneously using a small network topology would need the most efficient distributed data paths, due to high traffic areas being managed and capacity utilisation being upheld to ensure one server is not overworked.

Additionally, load balancing performs different functions such as only sending traffic to online servers to ensure availability and reliability of requests, which can occur from both client and network side of things. For example, request happen either end of the network, from the controller, switches, or hosts. Not all 250 employees will be logged on simultaneously, therefore, if all the servers were in use, capacity utilisation will be wasted, incurring higher costs.

However, load balancing provides a flexible service where servers can be added or subtracted anytime depending on demand levels, meaning expenses and capital of running the servers will decrease. (Creating manually – installing flows in flow table of switch)

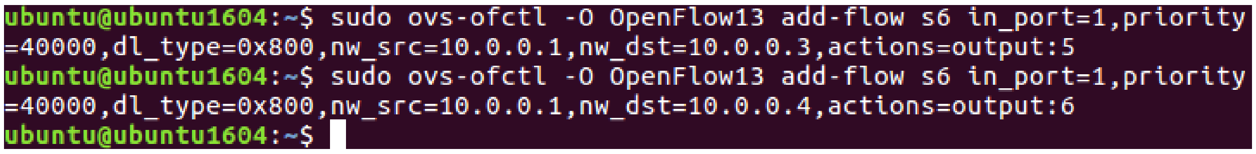


```
*** Results: 8% dropped (51/56 received)
mininet> pingall
*** Ping: testing ping reachability
h7 -> h5 h2 h8 h1 h6 h4 h3
h5 -> h7 h2 h8 h1 h6 h4 h3
h2 -> h7 h5 h8 h1 h6 h4 h3
h8 -> h7 h5 h2 X h6 h4 h3
h1 -> h7 h5 h2 X h6 h4 h3
h6 -> h7 h5 h2 h8 h1 h4 h3
h4 -> h7 h5 h2 h8 h1 h6 h3
h3 -> h7 h5 h2 h8 h1 h6 h4
*** Results: 3% dropped (54/56 received)
mininet> net
h7 h7-eth0:s6-eth1
h5 h5-eth0:s5-eth6
h2 h2-eth0:s6-eth3
h8 h8-eth0:s5-eth1
h1 h1-eth0:s6-eth2
h6 h6-eth0:s6-eth4
h4 h4-eth0:s5-eth5
h3 h3-eth0:s5-eth4
s1 lo:  s1-eth1:s6-eth5 s1-eth2:s3-eth1
s4 lo:  s4-eth1:s2-eth2 s4-eth2:s5-eth2
s3 lo:  s3-eth1:s1-eth2 s3-eth2:s5-eth3
s6 lo:  s6-eth1:h7-eth0 s6-eth2:h1-eth0 s6-eth3:h2-eth0 s6-eth4:h6-eth0 s6-eth5:s1-eth1 s6-eth6:s2-eth1
s5 lo:  s5-eth1:h8-eth0 s5-eth2:s4-eth2 s5-eth3:s3-eth2 s5-eth4:h3-eth0 s5-eth5:h4-eth0 s5-eth6:h5-eth0
s2 lo:  s2-eth1:s6-eth6 s2-eth2:s4-eth1
c0
mininet> 
```

Figure 16: Mininet Network

Figure 17: Adding Flows

Figure 16 shows a connection peered Mininet network, which is an overview from the topology as shown in figure 1. From this, connections can be identified in order to make load balancing changes to the network. For

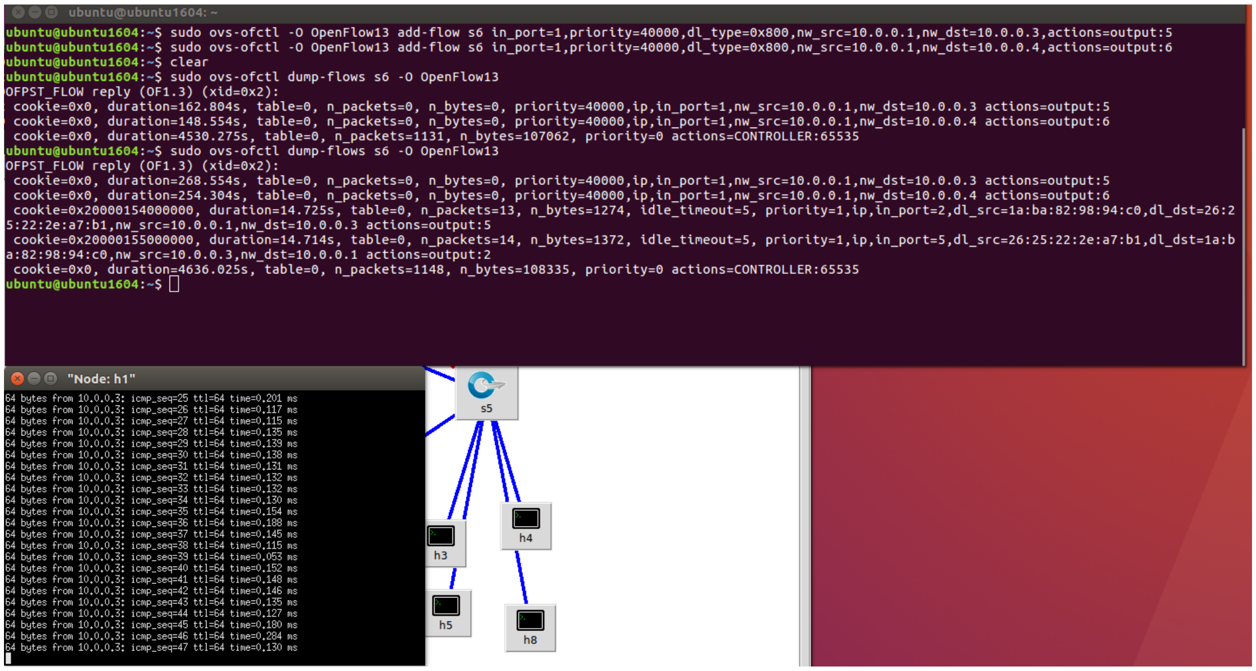


```
ubuntu@ubuntu1604:~$ sudo ovs-ofctl -O OpenFlow13 add-flow s6 in_port=1,priority
=40000,dl_type=0x800,nw_src=10.0.0.1,nw_dst=10.0.0.3,actions=output:5
ubuntu@ubuntu1604:~$ sudo ovs-ofctl -O OpenFlow13 add-flow s6 in_port=1,priority
=40000,dl_type=0x800,nw_src=10.0.0.1,nw_dst=10.0.0.4,actions=output:6
ubuntu@ubuntu1604:~$ 
```

example, all hosts only have 1 port, and they are all connected to port 0. Host 7 is connected to port 0 through switch 6 on port 1. The below switches show different ports since they have access to multiple.

Within the network topology (Figure 1), the load balancing will take place at switch 6, port 1, on host 3 port 5, and on host 4 port 6 as shown in Figure 17. From figure 16, you can see where switch 6 is connected to port 1, and where each of the different ports are connected from the above commands. For example, host 3 is connected to port 0, through switch 5 on port 4. This load balancing is taking place through switch 1, where switch 6 uses port 5, and on switch 2, uses port 6.

Installing the flow entries means we can see the load balancing take place through the number of packets going through each switch and each port.



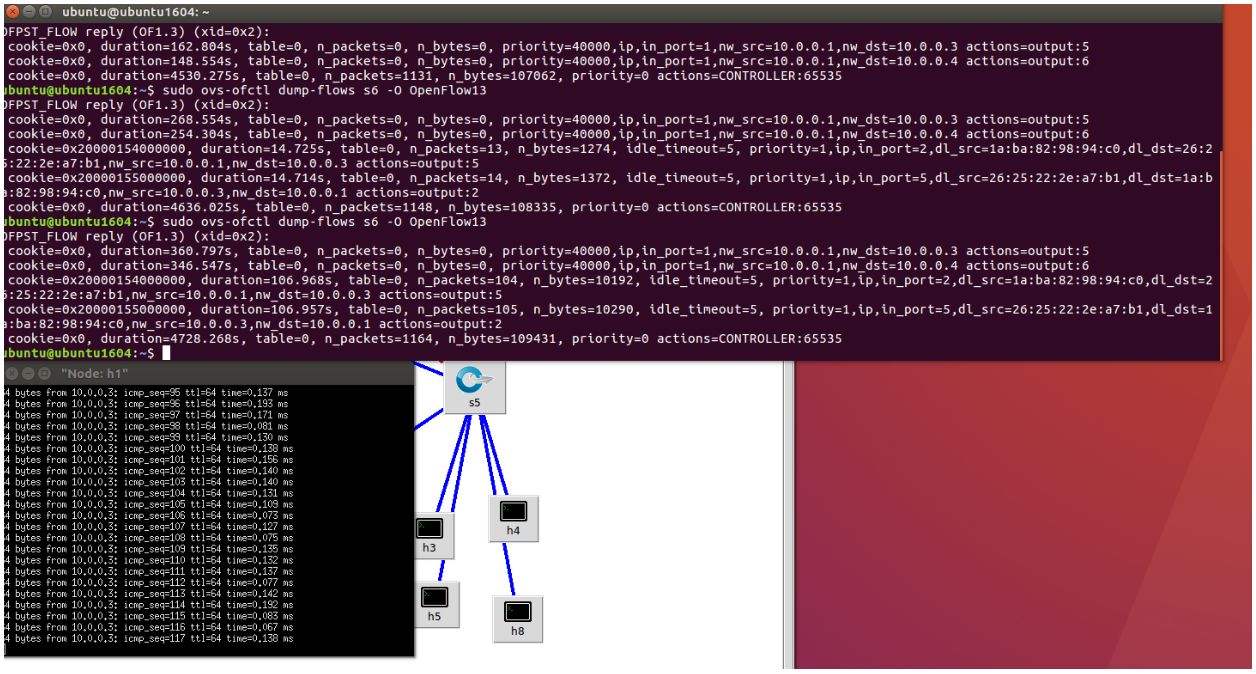
```
ubuntu@ubuntu1604:~$ sudo ovs-ofctl -O OpenFlow13 add-flow s6 in_port=1,priority=40000,dl_type=0x800,nw_src=10.0.0.1,nw_dst=10.0.0.3,actions=output:5
ubuntu@ubuntu1604:~$ sudo ovs-ofctl -O OpenFlow13 add-flow s6 in_port=1,priority=40000,dl_type=0x800,nw_src=10.0.0.1,nw_dst=10.0.0.4,actions=output:6
ubuntu@ubuntu1604:~$ clear
ubuntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=162.804s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x0, duration=148.554s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:6
 cookie=0x0, duration=4530.275s, table=0, n_packets=1131, n_bytes=107062, priority=0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=268.554s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x0, duration=254.304s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:6
 cookie=0x20000154000000, duration=14.725s, table=0, n_packets=13, n_bytes=1274, idle_timeout=5, priority=1,ip,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=26:2
5:22:2e:a7:b1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x20000155000000, duration=14.714s, table=0, n_packets=14, n_bytes=1372, idle_timeout=5, priority=1,ip,in_port=5,dl_src=26:25:22:2e:a7:b1,dl_dst=1a:b
a:82:98:94:c0,nw_src=10.0.0.3,nw_dst=10.0.0.1 actions=output:2
 cookie=0x0, duration=4636.025s, table=0, n_packets=1148, n_bytes=108335, priority=0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$ 
```

```
64 bytes from 10.0.0.3: icmp_seq=25 ttl=64 time=0.201 ms
64 bytes from 10.0.0.3: icmp_seq=26 ttl=64 time=0.117 ms
64 bytes from 10.0.0.3: icmp_seq=27 ttl=64 time=0.115 ms
64 bytes from 10.0.0.3: icmp_seq=28 ttl=64 time=0.135 ms
64 bytes from 10.0.0.3: icmp_seq=29 ttl=64 time=0.139 ms
64 bytes from 10.0.0.3: icmp_seq=30 ttl=64 time=0.138 ms
64 bytes from 10.0.0.3: icmp_seq=31 ttl=64 time=0.131 ms
64 bytes from 10.0.0.3: icmp_seq=32 ttl=64 time=0.132 ms
64 bytes from 10.0.0.3: icmp_seq=33 ttl=64 time=0.132 ms
64 bytes from 10.0.0.3: icmp_seq=34 ttl=64 time=0.130 ms
64 bytes from 10.0.0.3: icmp_seq=35 ttl=64 time=0.154 ms
64 bytes from 10.0.0.3: icmp_seq=36 ttl=64 time=0.188 ms
64 bytes from 10.0.0.3: icmp_seq=37 ttl=64 time=0.145 ms
64 bytes from 10.0.0.3: icmp_seq=38 ttl=64 time=0.115 ms
64 bytes from 10.0.0.3: icmp_seq=39 ttl=64 time=0.053 ms
64 bytes from 10.0.0.3: icmp_seq=40 ttl=64 time=0.152 ms
64 bytes from 10.0.0.3: icmp_seq=41 ttl=64 time=0.148 ms
64 bytes from 10.0.0.3: icmp_seq=42 ttl=64 time=0.146 ms
64 bytes from 10.0.0.3: icmp_seq=43 ttl=64 time=0.135 ms
64 bytes from 10.0.0.3: icmp_seq=44 ttl=64 time=0.127 ms
64 bytes from 10.0.0.3: icmp_seq=45 ttl=64 time=0.180 ms
64 bytes from 10.0.0.3: icmp_seq=46 ttl=64 time=0.284 ms
64 bytes from 10.0.0.3: icmp_seq=47 ttl=64 time=0.130 ms
```

Figure 18: Testing Packet Count

Figure 19: Packet Count Increasing

Now the flow entries are installed, the packet count can be traced through the switches and hosts to ensure load balancing is occurring. Using the command line node, host 1 is opened to ping host 3 continually. The Openflow dump

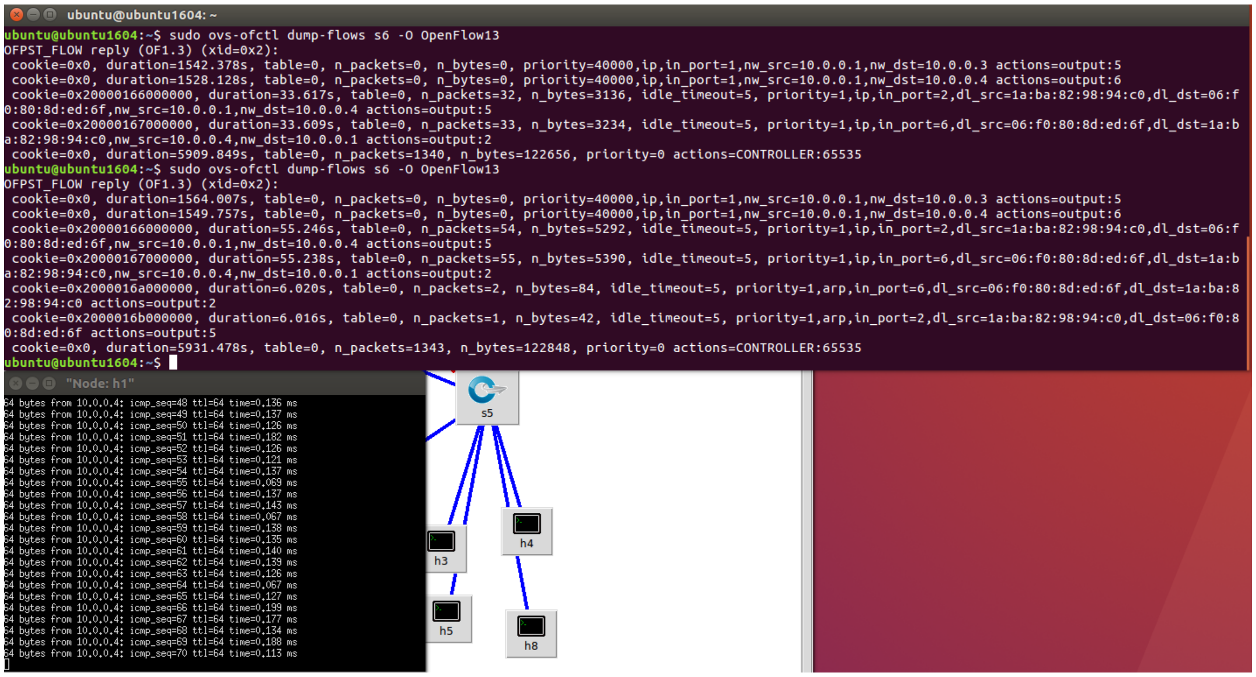


```
FPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=162.804s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x0, duration=148.554s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:6
 cookie=0x0, duration=4530.275s, table=0, n_packets=1131, n_bytes=107062, priority=0 actions=CONTROLLER:65535
buntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
FPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=268.554s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x0, duration=254.304s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:6
 cookie=0x20000154000000, duration=14.725s, table=0, n_packets=13, n_bytes=1274, idle_timeout=5, priority=1,ip,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=26:2
:22:2e:a7:b1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x20000155000000, duration=14.714s, table=0, n_packets=14, n_bytes=1372, idle_timeout=5, priority=1,ip,in_port=5,dl_src=26:25:22:2e:a7:b1,dl_dst=1a:b
:82:98:94:c0,nw_src=10.0.0.3,nw_dst=10.0.0.1 actions=output:2
 cookie=0x0, duration=4636.025s, table=0, n_packets=1148, n_bytes=108335, priority=0 actions=CONTROLLER:65535
buntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
FPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=360.797s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x0, duration=346.547s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:6
 cookie=0x20000154000000, duration=106.968s, table=0, n_packets=104, n_bytes=10192, idle_timeout=5, priority=1,ip,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=2
:25:22:2e:a7:b1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x20000155000000, duration=106.957s, table=0, n_packets=105, n_bytes=10290, idle_timeout=5, priority=1,ip,in_port=5,dl_src=26:25:22:2e:a7:b1,dl_dst=1
:ba:82:98:94:c0,nw_src=10.0.0.3,nw_dst=10.0.0.1 actions=output:2
 cookie=0x0, duration=4728.268s, table=0, n_packets=1164, n_bytes=109431, priority=0 actions=CONTROLLER:65535
buntu@ubuntu1604:~$ 
```

```
4 bytes from 10.0.0.3: icmp_seq=95 ttl=64 time=0.137 ms
4 bytes from 10.0.0.3: icmp_seq=96 ttl=64 time=0.193 ms
4 bytes from 10.0.0.3: icmp_seq=97 ttl=64 time=0.171 ms
4 bytes from 10.0.0.3: icmp_seq=98 ttl=64 time=0.081 ms
4 bytes from 10.0.0.3: icmp_seq=99 ttl=64 time=0.130 ms
4 bytes from 10.0.0.3: icmp_seq=100 ttl=64 time=0.138 ms
4 bytes from 10.0.0.3: icmp_seq=101 ttl=64 time=0.156 ms
4 bytes from 10.0.0.3: icmp_seq=102 ttl=64 time=0.140 ms
4 bytes from 10.0.0.3: icmp_seq=103 ttl=64 time=0.140 ms
4 bytes from 10.0.0.3: icmp_seq=104 ttl=64 time=0.131 ms
4 bytes from 10.0.0.3: icmp_seq=105 ttl=64 time=0.109 ms
4 bytes from 10.0.0.3: icmp_seq=106 ttl=64 time=0.073 ms
4 bytes from 10.0.0.3: icmp_seq=107 ttl=64 time=0.127 ms
4 bytes from 10.0.0.3: icmp_seq=108 ttl=64 time=0.075 ms
4 bytes from 10.0.0.3: icmp_seq=109 ttl=64 time=0.135 ms
4 bytes from 10.0.0.3: icmp_seq=110 ttl=64 time=0.132 ms
4 bytes from 10.0.0.3: icmp_seq=111 ttl=64 time=0.137 ms
4 bytes from 10.0.0.3: icmp_seq=112 ttl=64 time=0.077 ms
4 bytes from 10.0.0.3: icmp_seq=113 ttl=64 time=0.142 ms
4 bytes from 10.0.0.3: icmp_seq=114 ttl=64 time=0.192 ms
4 bytes from 10.0.0.3: icmp_seq=115 ttl=64 time=0.083 ms
4 bytes from 10.0.0.3: icmp_seq=116 ttl=64 time=0.067 ms
4 bytes from 10.0.0.3: icmp_seq=117 ttl=64 time=0.138 ms
```

commands as shown in Figure 18 shows the current packet count while the ping is being carried out, and this packet count increases to 13 on host 3 and 14 on host 1.

Continuing with the packet testing, Figure 19 shows the same commands being used from host 1 to host 3, except packet count is increasing without the 2nd link being utilized. As a result, as long as only one packet counter is increasing, load balancing is shown to be taking place.



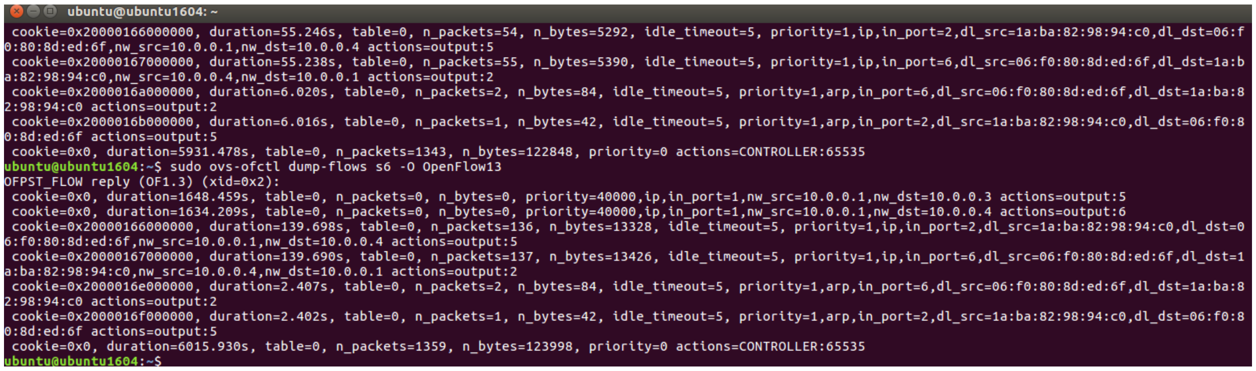
```
ubuntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=1542.378s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x0, duration=1528.128s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:6
 cookie=0x20000166000000, duration=33.617s, table=0, n_packets=32, n_bytes=3136, idle_timeout=5, priority=1,ip,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=06:f
0:80:8d:ed:6f,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:5
 cookie=0x20000167000000, duration=33.609s, table=0, n_packets=33, n_bytes=3234, idle_timeout=5, priority=1,ip,in_port=6,dl_src=06:f0:80:8d:ed:6f,dl_dst=1a:b
a:82:98:94:c0,nw_src=10.0.0.4,nw_dst=10.0.0.1 actions=output:2
 cookie=0x0, duration=5909.849s, table=0, n_packets=1340, n_bytes=122656, priority=0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=1564.007s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x0, duration=1549.757s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:6
 cookie=0x20000166000000, duration=55.246s, table=0, n_packets=54, n_bytes=5292, idle_timeout=5, priority=1,ip,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=06:f
0:80:8d:ed:6f,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:5
 cookie=0x20000167000000, duration=55.238s, table=0, n_packets=55, n_bytes=5390, idle_timeout=5, priority=1,ip,in_port=6,dl_src=06:f0:80:8d:ed:6f,dl_dst=1a:b
a:82:98:94:c0,nw_src=10.0.0.4,nw_dst=10.0.0.1 actions=output:2
 cookie=0x2000016a000000, duration=6.020s, table=0, n_packets=2, n_bytes=84, idle_timeout=5, priority=1,arp,in_port=6,dl_src=06:f0:80:8d:ed:6f,dl_dst=1a:ba:8
2:98:94:c0 actions=output:2
 cookie=0x2000016b000000, duration=6.016s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=06:f0:8
0:8d:ed:6f actions=output:5
 cookie=0x0, duration=5931.478s, table=0, n_packets=1343, n_bytes=122848, priority=0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$
```

```
64 bytes from 10.0.0.4: icmp_seq=48 ttl=64 time=0.136 ms
64 bytes from 10.0.0.4: icmp_seq=49 ttl=64 time=0.137 ms
64 bytes from 10.0.0.4: icmp_seq=50 ttl=64 time=0.126 ms
64 bytes from 10.0.0.4: icmp_seq=51 ttl=64 time=0.182 ms
64 bytes from 10.0.0.4: icmp_seq=52 ttl=64 time=0.126 ms
64 bytes from 10.0.0.4: icmp_seq=53 ttl=64 time=0.121 ms
64 bytes from 10.0.0.4: icmp_seq=54 ttl=64 time=0.137 ms
64 bytes from 10.0.0.4: icmp_seq=55 ttl=64 time=0.069 ms
64 bytes from 10.0.0.4: icmp_seq=56 ttl=64 time=0.137 ms
64 bytes from 10.0.0.4: icmp_seq=57 ttl=64 time=0.143 ms
64 bytes from 10.0.0.4: icmp_seq=58 ttl=64 time=0.067 ms
64 bytes from 10.0.0.4: icmp_seq=59 ttl=64 time=0.138 ms
64 bytes from 10.0.0.4: icmp_seq=60 ttl=64 time=0.135 ms
64 bytes from 10.0.0.4: icmp_seq=61 ttl=64 time=0.140 ms
64 bytes from 10.0.0.4: icmp_seq=62 ttl=64 time=0.139 ms
64 bytes from 10.0.0.4: icmp_seq=63 ttl=64 time=0.126 ms
64 bytes from 10.0.0.4: icmp_seq=64 ttl=64 time=0.067 ms
64 bytes from 10.0.0.4: icmp_seq=65 ttl=64 time=0.127 ms
64 bytes from 10.0.0.4: icmp_seq=66 ttl=64 time=0.199 ms
64 bytes from 10.0.0.4: icmp_seq=67 ttl=64 time=0.177 ms
64 bytes from 10.0.0.4: icmp_seq=68 ttl=64 time=0.134 ms
64 bytes from 10.0.0.4: icmp_seq=69 ttl=64 time=0.188 ms
64 bytes from 10.0.0.4: icmp_seq=70 ttl=64 time=0.113 ms
```

Figure 20: Packet Count from H1 to H4

Figure 21: Increasing Packet Amounts

To ensure it works correctly across all hosts, host 4



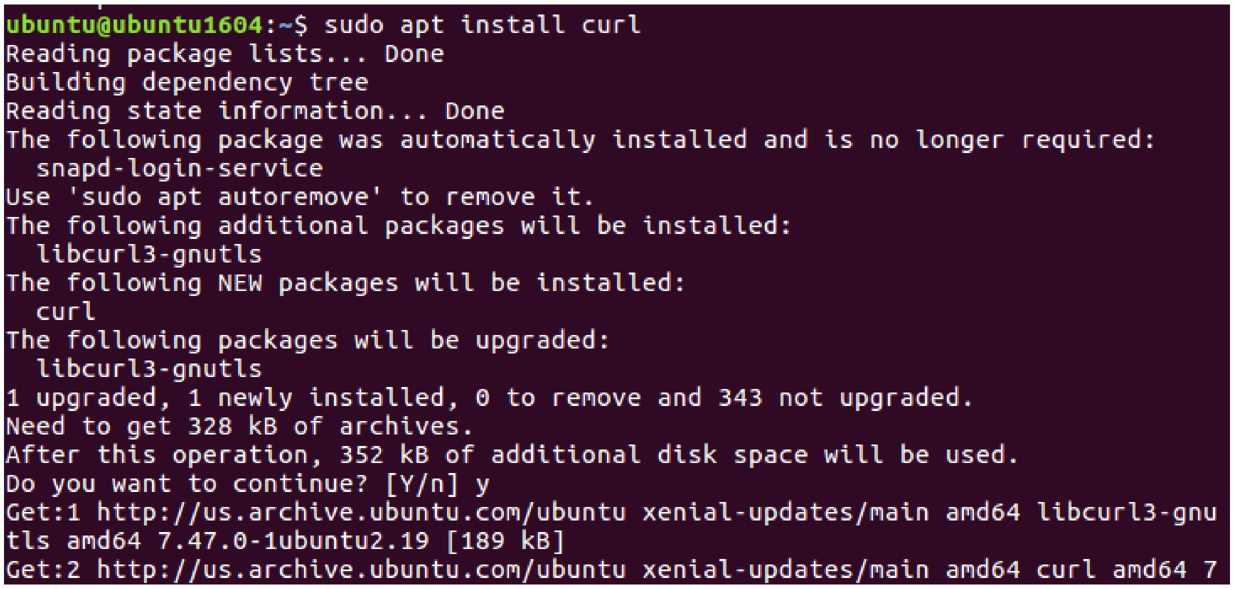
```
 cookie=0x20000166000000, duration=55.246s, table=0, n_packets=54, n_bytes=5292, idle_timeout=5, priority=1,ip,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=06:f
0:80:8d:ed:6f,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:5
 cookie=0x20000167000000, duration=55.238s, table=0, n_packets=55, n_bytes=5390, idle_timeout=5, priority=1,ip,in_port=6,dl_src=06:f0:80:8d:ed:6f,dl_dst=1a:b
a:82:98:94:c0,nw_src=10.0.0.4,nw_dst=10.0.0.1 actions=output:2
 cookie=0x2000016a000000, duration=6.020s, table=0, n_packets=2, n_bytes=84, idle_timeout=5, priority=1,arp,in_port=6,dl_src=06:f0:80:8d:ed:6f,dl_dst=1a:ba:8
2:98:94:c0 actions=output:2
 cookie=0x2000016b000000, duration=6.016s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=06:f0:8
0:8d:ed:6f actions=output:5
 cookie=0x0, duration=5931.478s, table=0, n_packets=1343, n_bytes=122848, priority=0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$ sudo ovs-ofctl dump-flows s6 -O OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=1648.459s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.3 actions=output:5
 cookie=0x0, duration=1634.209s, table=0, n_packets=0, n_bytes=0, priority=40000,ip,in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:6
 cookie=0x20000166000000, duration=139.698s, table=0, n_packets=136, n_bytes=13328, idle_timeout=5, priority=1,ip,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=0
6:f0:80:8d:ed:6f,nw_src=10.0.0.1,nw_dst=10.0.0.4 actions=output:5
 cookie=0x20000167000000, duration=139.690s, table=0, n_packets=137, n_bytes=13426, idle_timeout=5, priority=1,ip,in_port=6,dl_src=06:f0:80:8d:ed:6f,dl_dst=1
a:ba:82:98:94:c0,nw_src=10.0.0.4,nw_dst=10.0.0.1 actions=output:2
 cookie=0x2000016e000000, duration=2.407s, table=0, n_packets=2, n_bytes=84, idle_timeout=5, priority=1,arp,in_port=6,dl_src=06:f0:80:8d:ed:6f,dl_dst=1a:ba:8
2:98:94:c0 actions=output:2
 cookie=0x2000016f000000, duration=2.402s, table=0, n_packets=1, n_bytes=42, idle_timeout=5, priority=1,arp,in_port=2,dl_src=1a:ba:82:98:94:c0,dl_dst=06:f0:8
0:8d:ed:6f actions=output:5
 cookie=0x0, duration=6015.930s, table=0, n_packets=1359, n_bytes=123998, priority=0 actions=CONTROLLER:65535
ubuntu@ubuntu1604:~$
```

was also tested as shown in Figure 20. From the dump flow tables, packets were increasing across the switches specified for load balancing, which shows the load balancing works correctly.

Figure 21 shows load balancing working with the packet count increasing. The controller is bidirectional, meaning doing the load balancing manually on one side results in the controller automatically load balancing the other.

3.3. Network Monitoring

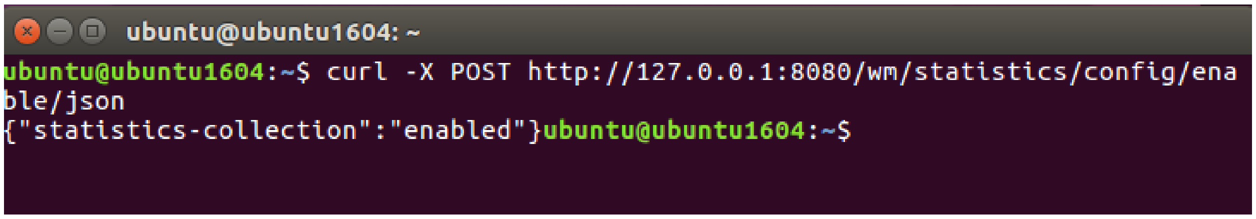
The last step of the network setup is to monitor the network to ensure all the controllers, switches and Openflow devices work correctly. Additionally, any statistics that a company want to find out can be viewed via the Floodlight rest API.



```
ubuntu@ubuntu1604:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libcurl3-gnutls
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl3-gnutls
1 upgraded, 1 newly installed, 0 to remove and 343 not upgraded.
Need to get 328 kB of archives.
After this operation, 352 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libcurl3-gnu
tls amd64 7.47.0-1ubuntu2.19 [189 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 curl amd64 7
```

Figure 22: Curl Installation

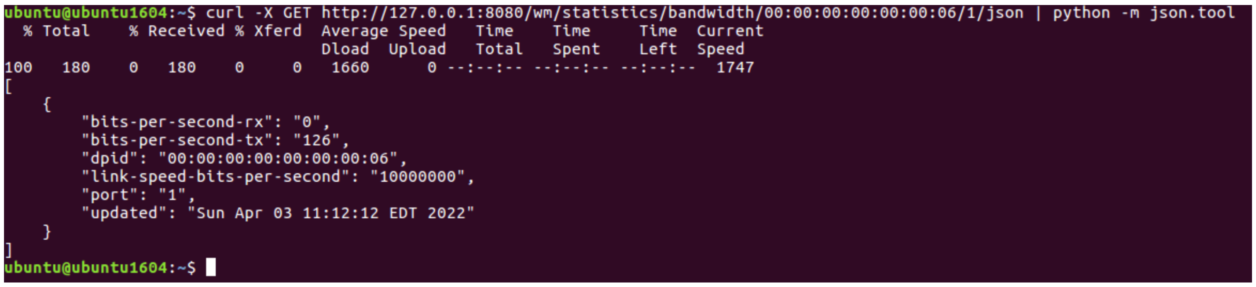
In order to access the rest API, curl needed to be installed (Figure 22), which is used to transfer data from different servers. Alternatively, python could be used, however, using the command line interface provide ease, rather than creating a new script.



```
ubuntu@ubuntu1604:~$ curl -X POST http://127.0.0.1:8080/wm/statistics/config/ena
ble/json
{"statistics-collection":"enabled"}ubuntu@ubuntu1604:~$
```

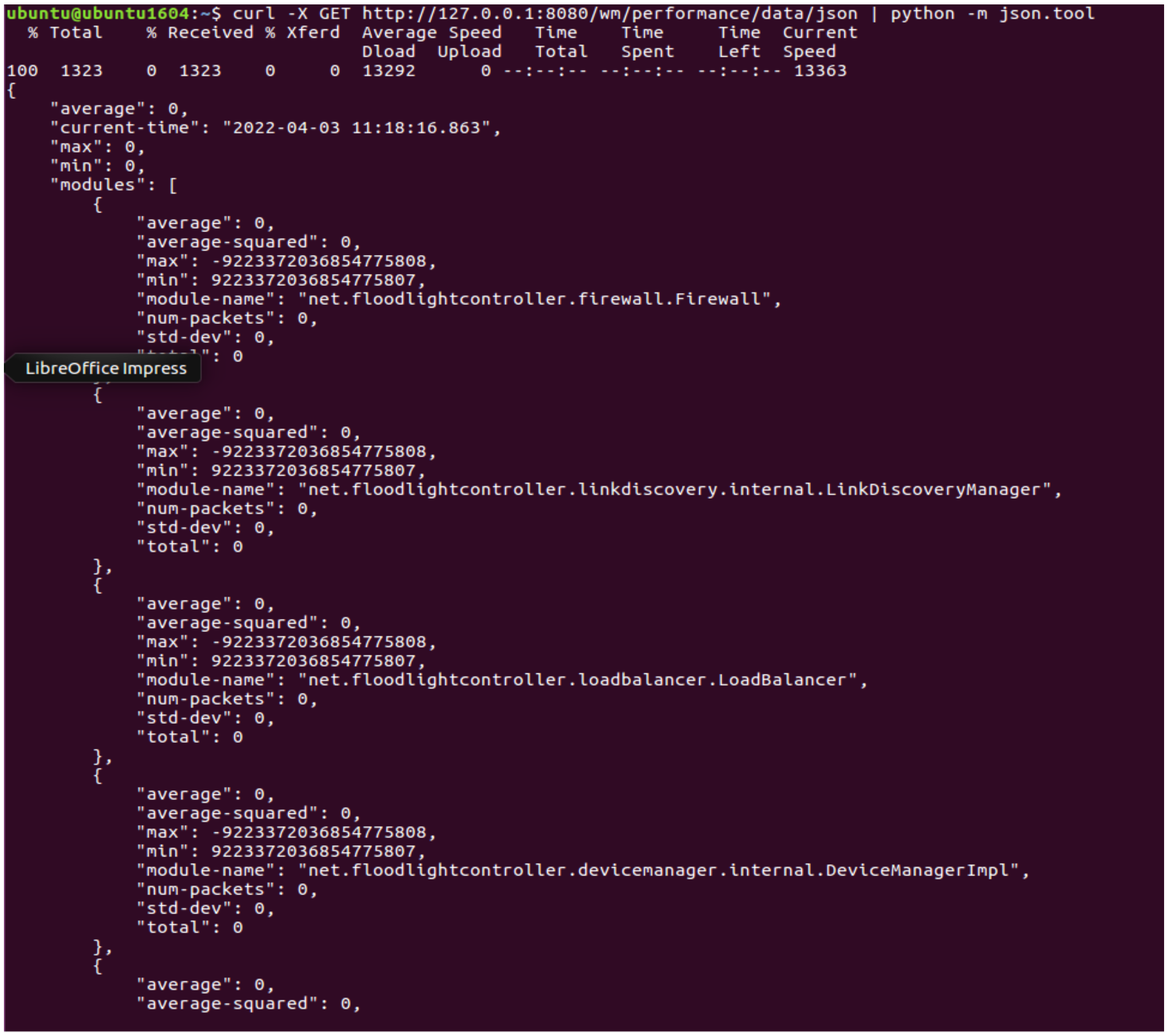
Figure 23: Enable Statistics Collection

To collect statistics, the controller has to be enabled through command in Figure 23. 127.0.0.1 is the default loopback address for the controller, and the port 8080 is the default API on the controller.



```
ubuntu@ubuntu1604:~$ curl -X GET http://127.0.0.1:8080/wm/statistics/bandwidth/00:00:00:00:00:00:00:06/1/json | python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   180    0   180    0     0   1660      0 --:--:-- --:--:-- --:--:--  1747
[
    {
        "bits-per-second-rx": "0",
        "bits-per-second-tx": "126",
        "dpid": "00:00:00:00:00:00:00:06",
        "link-speed-bits-per-second": "10000000",
        "port": "1",
        "updated": "Sun Apr 03 11:12:12 EDT 2022"
    }
]
ubuntu@ubuntu1604:~$ 
```

Figure 24: Bandwidth Statistics



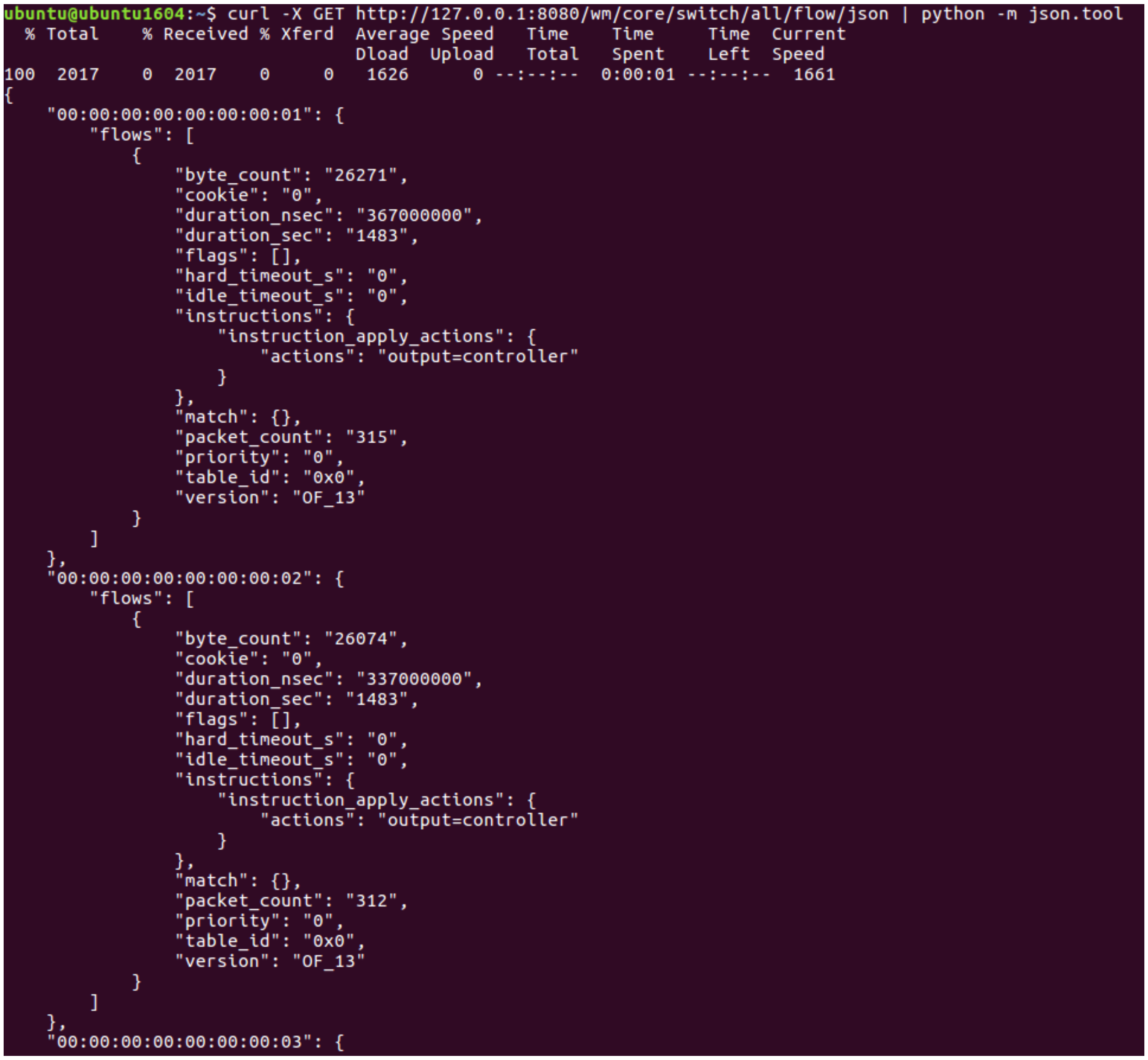
```
ubuntu@ubuntu1604:~$ curl -X GET http://127.0.0.1:8080/wm/performance/data/json | python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1323    0  1323    0     0  13292      0 --:--:-- --:--:-- --:--:-- 13363
{
    "average": 0,
    "current-time": "2022-04-03 11:18:16.863",
    "max": 0,
    "min": 0,
    "modules": [
        {
            "average": 0,
            "average-squared": 0,
            "max": -9223372036854775808,
            "min": 9223372036854775807,
            "module-name": "net.floodlightcontroller.firewall.Firewall",
            "num-packets": 0,
            "std-dev": 0,
            "total": 0
        },
        {
            "average": 0,
            "average-squared": 0,
            "max": -9223372036854775808,
            "min": 9223372036854775807,
            "module-name": "net.floodlightcontroller.linkdiscovery.internal.LinkDiscoveryManager",
            "num-packets": 0,
            "std-dev": 0,
            "total": 0
        },
        {
            "average": 0,
            "average-squared": 0,
            "max": -9223372036854775808,
            "min": 9223372036854775807,
            "module-name": "net.floodlightcontroller.loadbalancer.LoadBalancer",
            "num-packets": 0,
            "std-dev": 0,
            "total": 0
        },
        {
            "average": 0,
            "average-squared": 0,
            "max": -9223372036854775808,
            "min": 9223372036854775807,
            "module-name": "net.floodlightcontroller.devicemanager.internal.DeviceManagerImpl",
            "num-packets": 0,
            "std-dev": 0,
            "total": 0
        },
        {
            "average": 0,
            "average-squared": 0,
```

Figure 25: Controller performance

The first statistic is bandwidth, which is important to understand the speeds of a network. The command was a part of the Floodlight rest API as seen above and the format is json because data results are more organised. As seen from Figure 24, the statistics display bits sent/received, DPID (switch ID), link speed and the port being queried.

Figure 26: Switch Flow Statistics

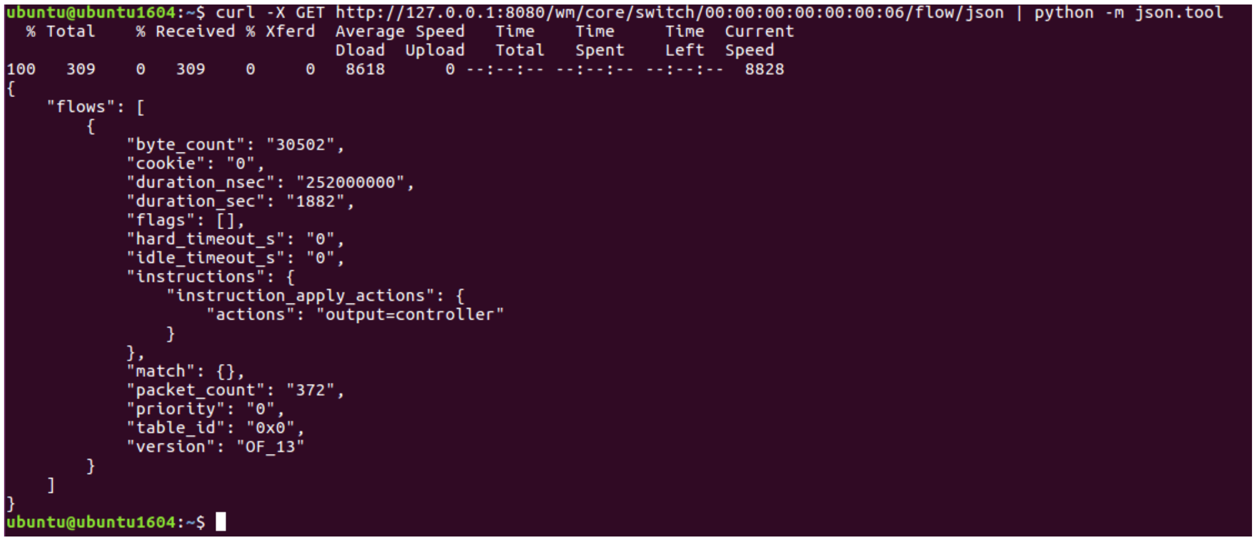
Figure 25 shows the performance of the controller in terms of average packet in processing



```
ubuntu@ubuntu1604:~$ curl -X GET http://127.0.0.1:8080/wm/core/switch/all/flow/json | python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2017    0  2017    0     0   1626      0 --:--:--  0:00:01 --:--:--  1661
{
    "00:00:00:00:00:00:00:01": {
        "flows": [
            {
                "byte_count": "26271",
                "cookie": "0",
                "duration_nsec": "367000000",
                "duration_sec": "1483",
                "flags": [],
                "hard_timeout_s": "0",
                "idle_timeout_s": "0",
                "instructions": {
                    "instruction_apply_actions": {
                        "actions": "output=controller"
                    }
                },
                "match": {},
                "packet_count": "315",
                "priority": "0",
                "table_id": "0x0",
                "version": "OF_13"
            }
        ]
    },
    "00:00:00:00:00:00:00:02": {
        "flows": [
            {
                "byte_count": "26074",
                "cookie": "0",
                "duration_nsec": "337000000",
                "duration_sec": "1483",
                "flags": [],
                "hard_timeout_s": "0",
                "idle_timeout_s": "0",
                "instructions": {
                    "instruction_apply_actions": {
                        "actions": "output=controller"
                    }
                },
                "match": {},
                "packet_count": "312",
                "priority": "0",
                "table_id": "0x0",
                "version": "OF_13"
            }
        ]
    },
    "00:00:00:00:00:00:00:03": {
```

times, which can be significant for a company when analysing performance of the controller which controls the entire network.

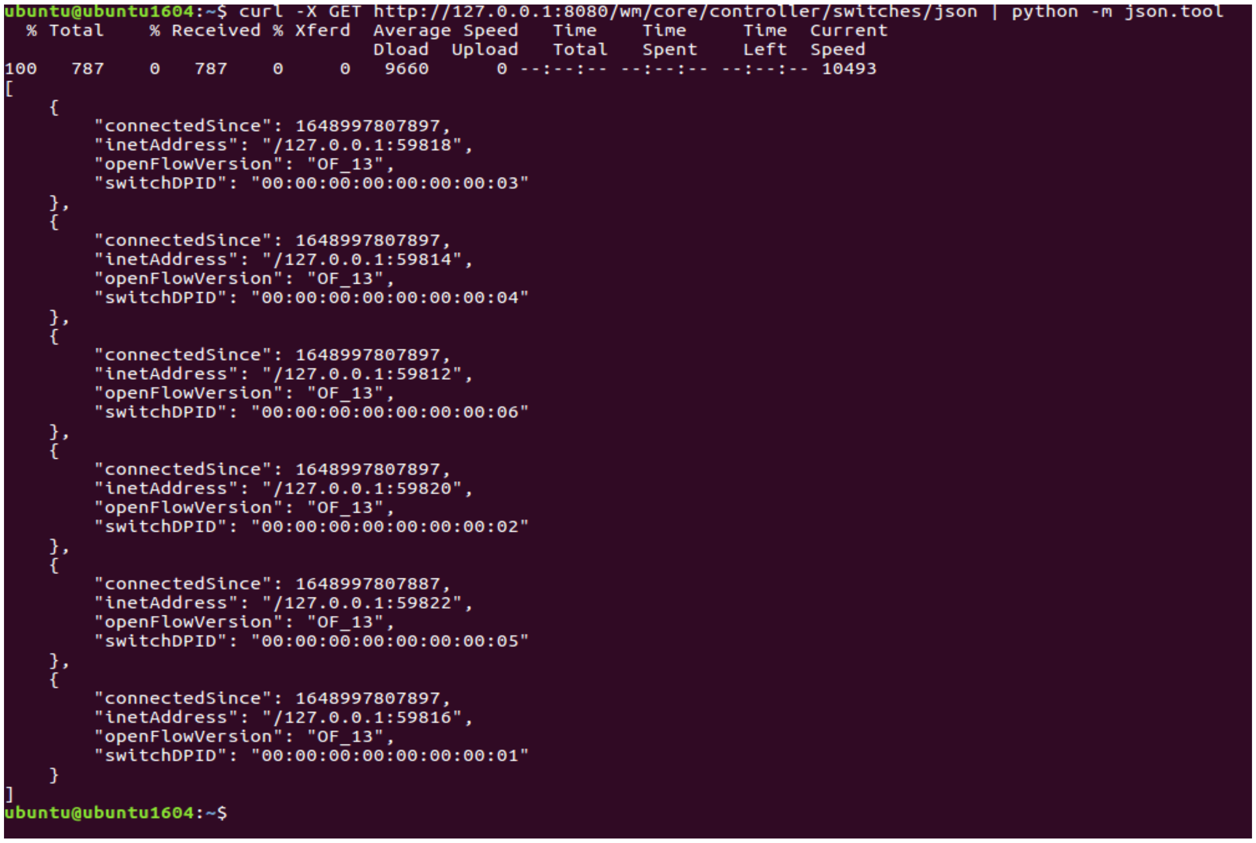
Figure 26 portrays the status of all switches, including packet counts, flows and table IDs. Using this can help a company establish switch performance if any issues arise.



```
ubuntu@ubuntu1604:~$ curl -X GET http://127.0.0.1:8080/wm/core/switch/00:00:00:00:00:00:00:06/flow/json | python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   309    0   309    0     0   8618      0 --:--:-- --:--:-- --:--:--  8828
{
    "flows": [
        {
            "byte_count": "30502",
            "cookie": "0",
            "duration_nsec": "252000000",
            "duration_sec": "1882",
            "flags": [],
            "hard_timeout_s": "0",
            "idle_timeout_s": "0",
            "instructions": {
                "instruction_apply_actions": {
                    "actions": "output=controller"
                }
            },
            "match": {},
            "packet_count": "372",
            "priority": "0",
            "table_id": "0x0",
            "version": "OF_13"
        }
    ]
}
ubuntu@ubuntu1604:~$
```

Figure 27: Individual Switch Flow Statistics

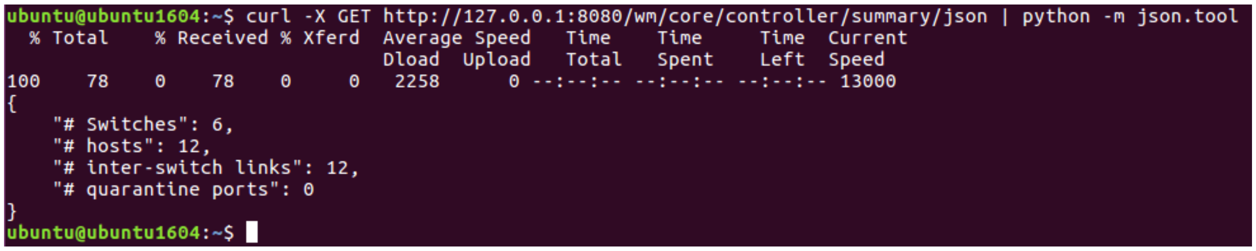
Figure 28: Controller Switches



```
ubuntu@ubuntu1604:~$ curl -X GET http://127.0.0.1:8080/wm/core/controller/switches/json | python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   787    0   787    0     0   9660      0 --:--:-- --:--:-- --:--:-- 10493
[
    {
        "connectedSince": 1648997807897,
        "inetAddress": "/127.0.0.1:59818",
        "openFlowVersion": "OF_13",
        "switchDPID": "00:00:00:00:00:00:00:03"
    },
    {
        "connectedSince": 1648997807897,
        "inetAddress": "/127.0.0.1:59814",
        "openFlowVersion": "OF_13",
        "switchDPID": "00:00:00:00:00:00:00:04"
    },
    {
        "connectedSince": 1648997807897,
        "inetAddress": "/127.0.0.1:59812",
        "openFlowVersion": "OF_13",
        "switchDPID": "00:00:00:00:00:00:00:06"
    },
    {
        "connectedSince": 1648997807897,
        "inetAddress": "/127.0.0.1:59820",
        "openFlowVersion": "OF_13",
        "switchDPID": "00:00:00:00:00:00:00:02"
    },
    {
        "connectedSince": 1648997807887,
        "inetAddress": "/127.0.0.1:59822",
        "openFlowVersion": "OF_13",
        "switchDPID": "00:00:00:00:00:00:00:05"
    },
    {
        "connectedSince": 1648997807897,
        "inetAddress": "/127.0.0.1:59816",
        "openFlowVersion": "OF_13",
        "switchDPID": "00:00:00:00:00:00:00:01"
    }
]
ubuntu@ubuntu1604:~$
```

Figure 27 shows a similar command switch, however, if a company wanted individual detailed statistics on a specific switch, for example, switch 6, then the above command finds those statistics.

Since switches are an important part of the network, the statistic as shown in Figure 28 portrays all of the switch DPIDs which are connected to the controller.



```
ubuntu@ubuntu1604:~$ curl -X GET http://127.0.0.1:8080/wm/core/controller/summary/json | python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    78    0    78    0     0   2258      0 --:--:-- --:--:-- --:--:-- 13000
{
    "# Switches": 6,
    "# hosts": 12,
    "# inter-switch links": 12,
    "# quarantine ports": 0
}
ubuntu@ubuntu1604:~$
```

Figure 29: Controller Link Summary

The final controller API statistic shows an overall network overview as shown in Figure 29, where all switches/hosts can be seen as an overview. This means that any new added network devices can be seen through one statistic command.