CS 6301 –Software defined Networks

Split the bandwidth Using QoS Policing on OpenDaylight

Mentor: Tim Hawks

Team Members:

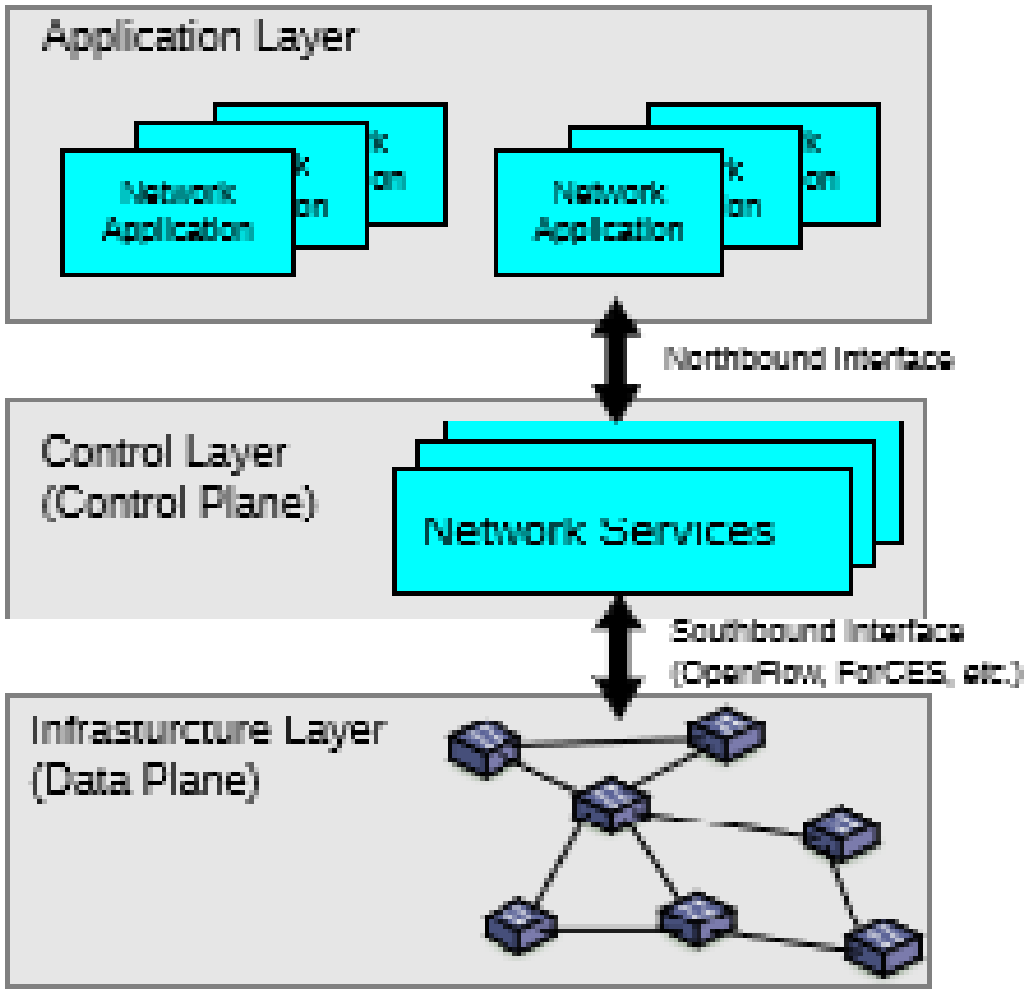
Sriharsha Ganja
Raghunandan N. R.
Abhinav Kumar Parakh
Arnav Sharma
Panchami Rudrakshi
Mounica Reddy Bojja

Outline

- Introduction
- Problem Definition
- Objective
- Setting up the environment
- Flow chart
- Working
- Results
- Conclusion

Introduction

What is SDN ?



Problem Definition & Solution

- Real time applications require stringent quality of service(QoS) guarantees.
- Thus there is a need for the network programmers to design network protocols that can deliver performance guarantees.

Solution:

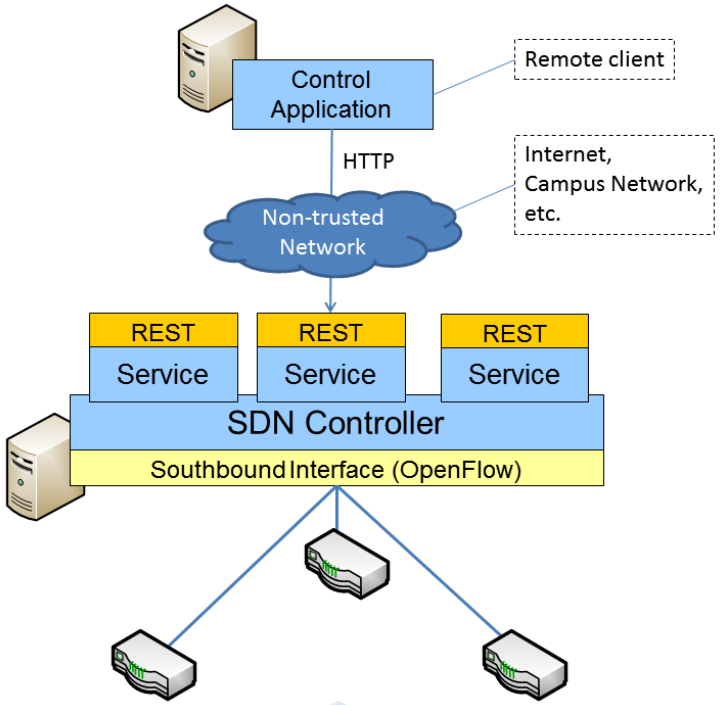
- Firstly, we define a QoS Management and Orchestration architecture that allows us to manage the network in a modular way.
- Secondly, we provide a seamless integration between the architecture and the standard SDN paradigm following the separation between the control and data planes.

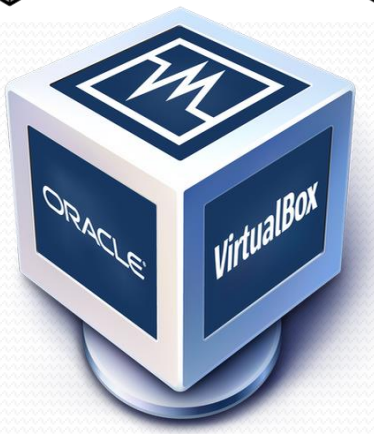
Objective

- Split the bandwidth according to the number of nodes connected.
- The system has several virtual machines running , connected to the SDN controller.
- **Input to the controller**: Sample of certain bandwidth.
- **Process**: Controller senses the input and identifies the bandwidth. According to the number of nodes/switches connected to the controller , the bandwidth is split.
- **Output**: Once the bandwidth reaches the nodes, it is tested using netperf

Setting up the environment

- SDN Controller (Open Daylight)
- Oracle VirtualBox
- Mininet





APPLICATION

TOPOLOGY,
NODES,HOSTS

APP

CREATE
TOPOLOGY

MININET

CREATE Meter
Max bw / no.of Hosts

Assign meter to
each flow

ODL
CONTROLLER

To create an meter with the band type drop which limits the bandwidth to 50000kbps you can use the following REST call:

```
PUT http://ip:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/meter/1
```

XML:

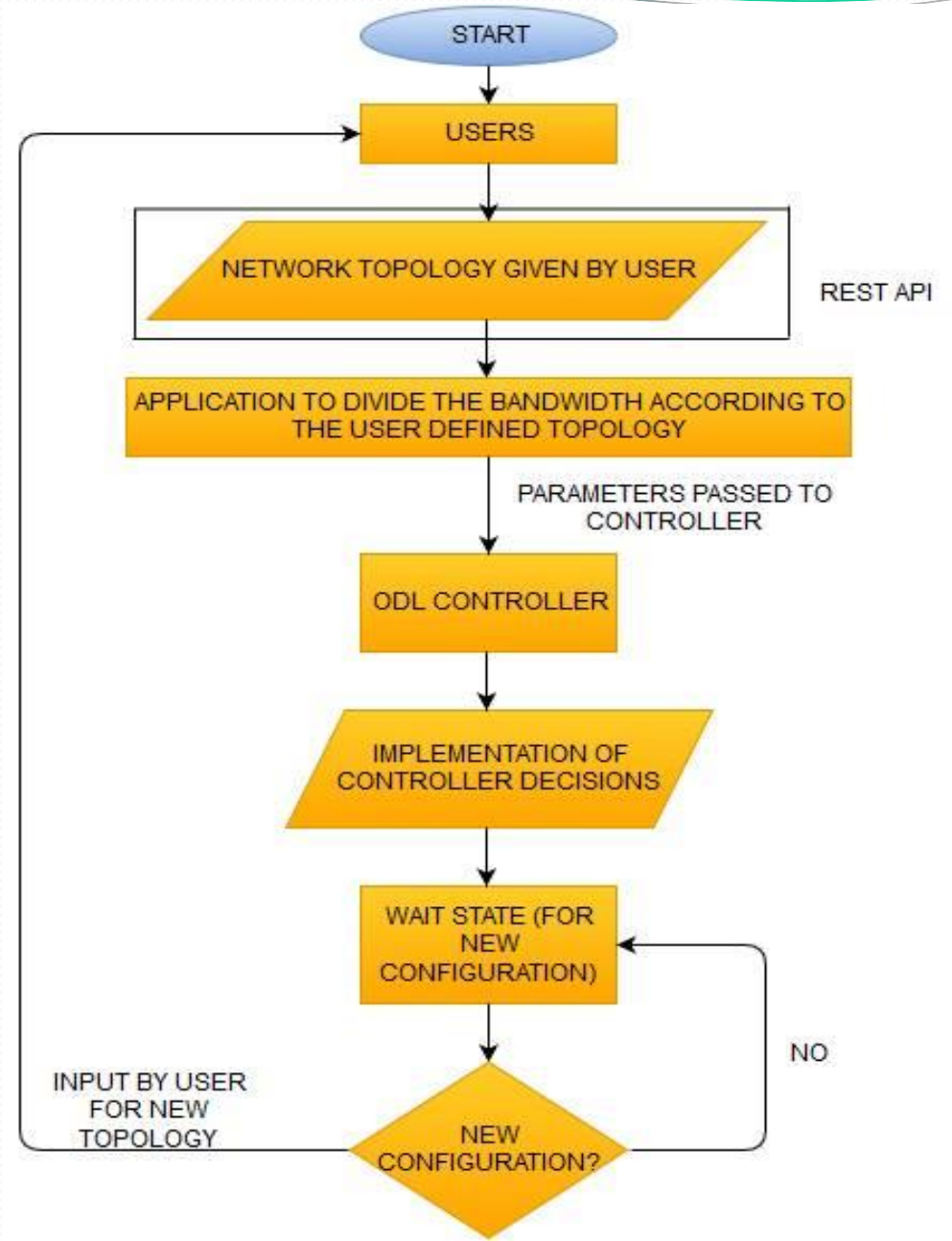
```
<meter
    xmlns="urn:opendaylight:flow:inventory">
    <meter-id>1</meter-id>
    <container-name>mymeter</container-name>
    <meter-name>mymeter</meter-name>
    <flags>meter-kbps</flags>
    <meter-band-headers>
        <meter-band-header>
            <band-id>0</band-id>
            <band-rate>50000</band-rate>
            <meter-band-types>
                <flags>ofpmbt-drop</flags>
            </meter-band-types>
            <band-burst-size>0</band-burst-size>
            <drop-rate>50000</drop-rate>
            <drop-burst-size>0</drop-burst-size>
        </meter-band-header>
    </meter-band-headers>
</meter>
```

To assign a meter with id 1 to a flow (which should be limited to band of the meter) with id 10

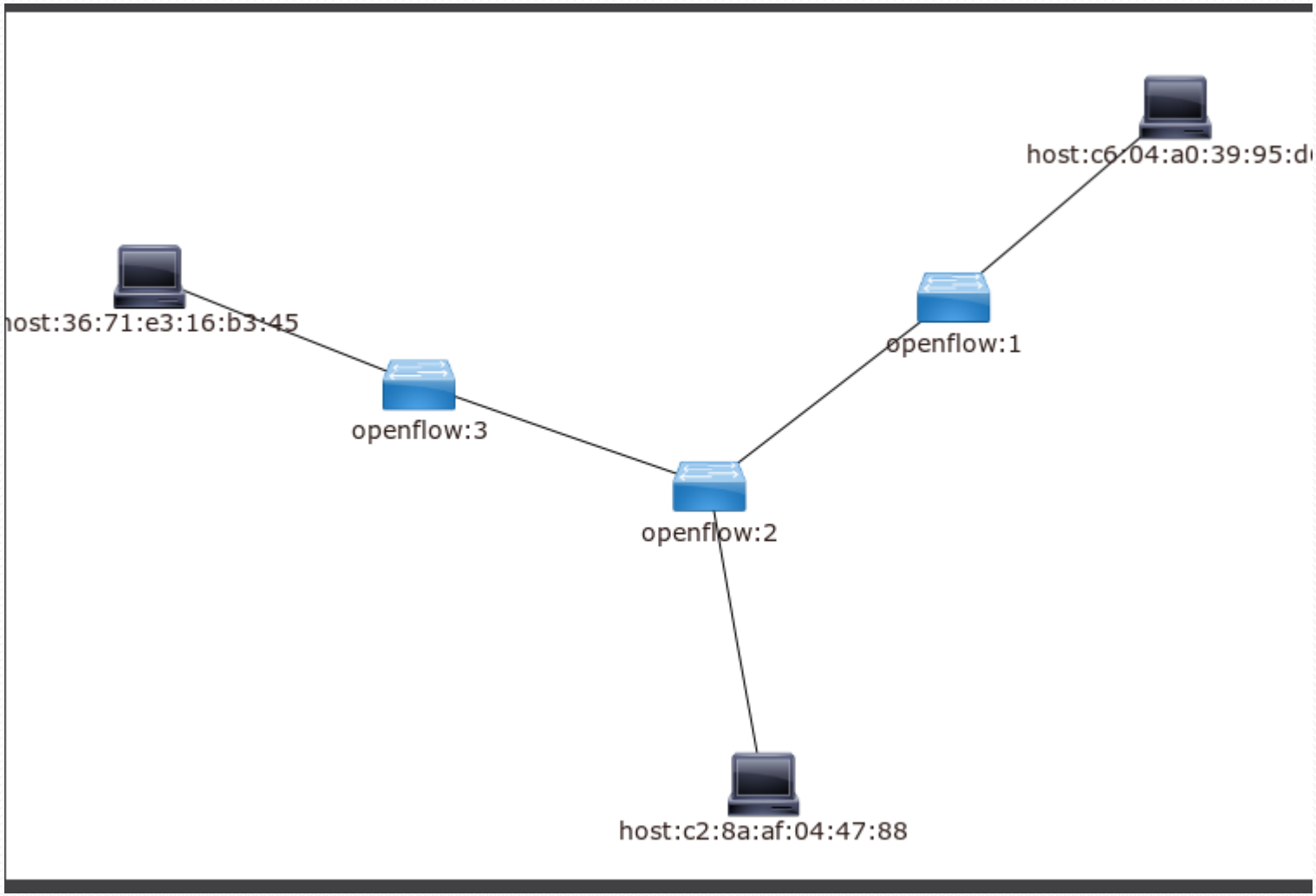
Address:

```
PUT http://ip:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/100/flow/10
```

Flow Chart

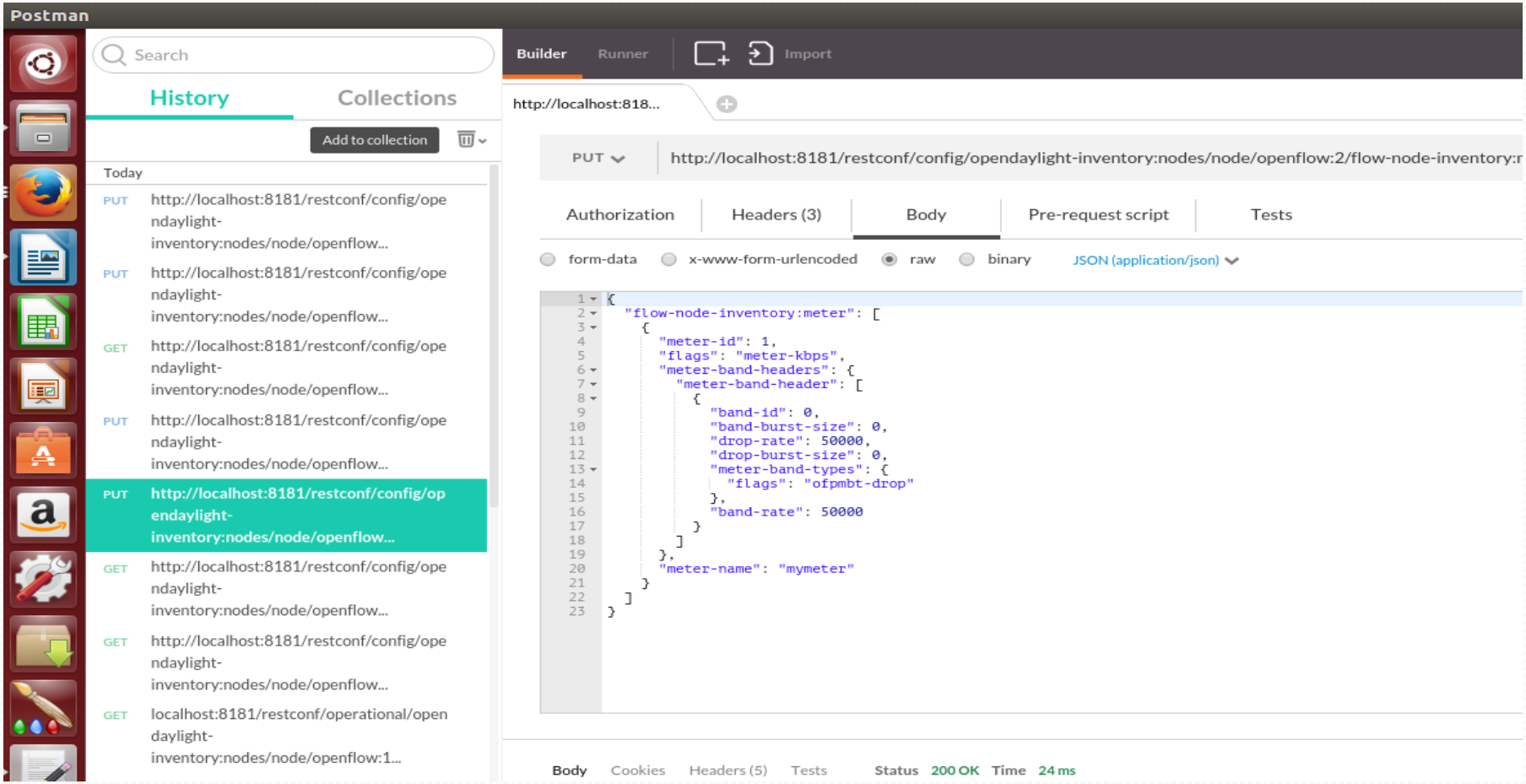

START
USERS
NETWORK TOPOLOGY GIVEN BY USER
REST API
APPLICATION TO DIVIDE THE BANDWIDTH ACCORDING TO THE USER DEFINED TOPOLOGY
PARAMETERS PASSED TO CONTROLLER
ODL CONTROLLER
IMPLEMENTATION OF CONTROLLER DECISIONS
WAIT STATE (FOR NEW CONFIGURATION)
NO
INPUT BY USER FOR NEW TOPOLOGY
NEW CONFIGURATION?

TOPOLOGY

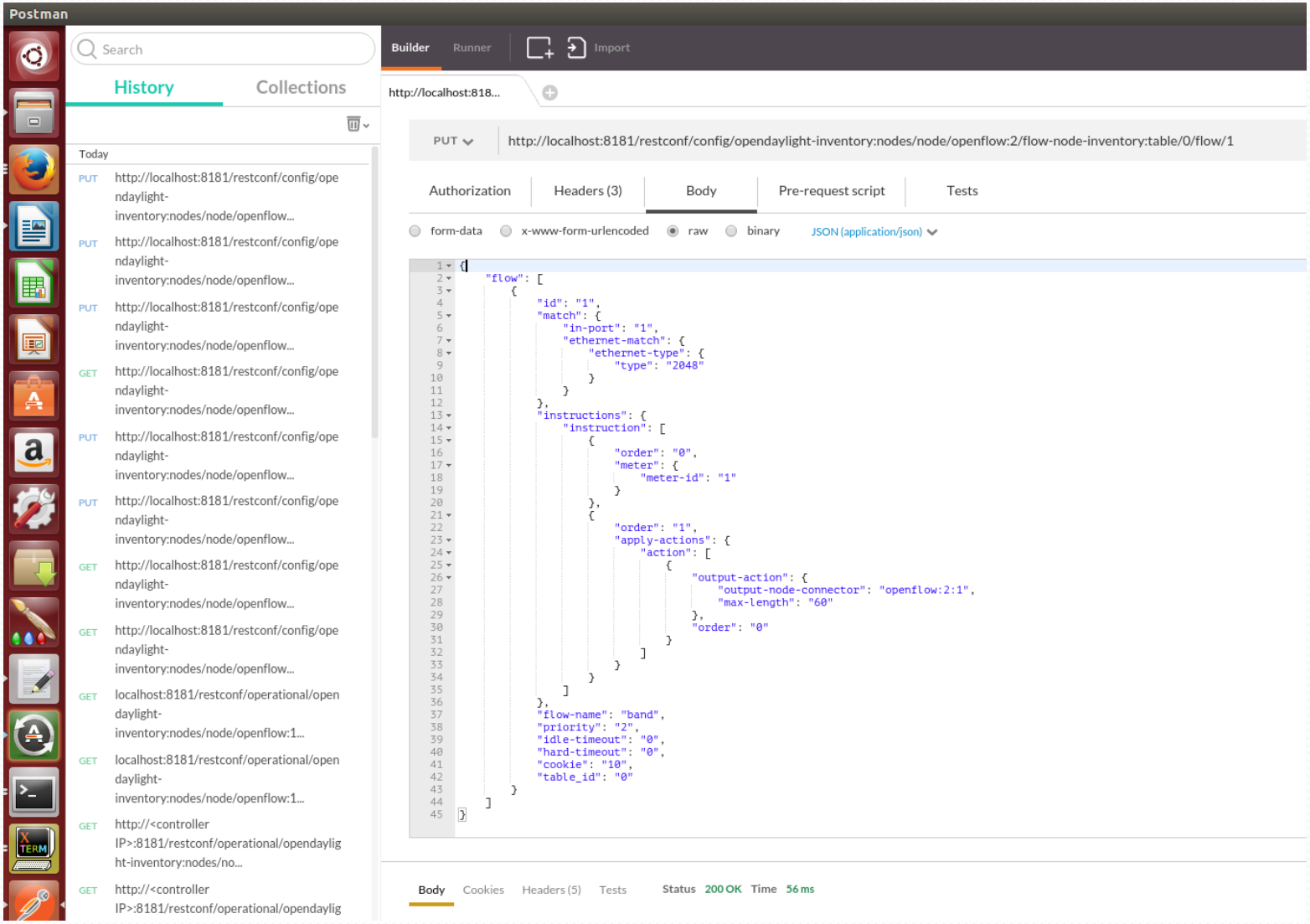

host:c6:04:a0:39:95:d
host:36:71:e3:16:b3:45
openflow:1
openflow:3
openflow:2
host:c2:8a:af:04:47:88

Working

Here we create meter-table,which is shown below.The meter-table triggers a variety of performance-related actions on a flow. Meter table consists of meter entries defining per flow meters. Per flow meters openflow to implement various simple Qos operations.



Here we are assigning the meter bandwidth to the flow



The flow assigned to the switch is checked here.

```
 cookie=0xa, duration=24.196s, table=0, n_packets=0, n_bytes=0, idle_age=24, priority=2,in_port=1,dl_dst=0e:a0:0e:ff:30:d2 actions=output:1
mininet> sh ovs-ofctl dump-flows s2
NXST_FLOW reply (xid=0x4):
 cookie=0x2b0000000000000b, duration=6280.799s, table=0, n_packets=1638, n_bytes=313684, idle_age=44, priority=2,in_port=3 actions=output:1,output:2
 cookie=0x2b00000000000009, duration=6280.799s, table=0, n_packets=3087185, n_bytes=2730777682, idle_age=65, priority=2,in_port=1 actions=output:2,output:3,CONTROLLER:65535
 cookie=0x2b0000000000000a, duration=6280.799s, table=0, n_packets=4181650, n_bytes=5418730559, idle_age=43, priority=2,in_port=2 actions=output:1,output:3
 cookie=0x2b00000000000009, duration=6286.746s, table=0, n_packets=2515, n_bytes=213775, idle_age=3, priority=100,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x2b00000000000009, duration=6286.736s, table=0, n_packets=50, n_bytes=7747, idle_age=6277, priority=0 actions=drop
 cookie=0xa, duration=89.953s, table=0, n_packets=0, n_bytes=0, idle_age=89, hard_age=4, priority=2,in_port=1,dl_dst=0e:a0:0e:ff:30:d2 actions=output:1
mininet>
```

Challenges Faced

We have passed the flows which is accepted by ODL controller, but ODL is not implementing them properly.

So to cross check we have done the following statistics:

1) Configuration Statistics

2) Operational Statistics

When we tried adding the flows, they were actually present in configuration statistics and not in operational statistics.

They are also present in the switch , which got the flows.

Conclusion

- —The purpose of this project was to create Qos based bandwidth split.
- —We were able to develop a SDN application to get the input of the number of nodes and assign bandwidth accordingly.
- We were able to create the meter table to measure the performance related issuesEndFragment

References

- http://nrlweb.cs.ucla.edu/publication/download/807/Enhancing_Quality_of_Service_in_Software-Defined_Networks_-_thesis_Francesco_Ongaro.pdf
- https://ask.opendaylight.org/question/2094/meter-bandwidth/
- https://ask.opendaylight.org/question/2094/meter-bandwidth/
- https://ask.opendaylight.org/question/4143/restconf-troubleshooting/?answer=4281#post-id-4281
- http://keepingitclassless.net/2014/07/sdn-protocols-2-openflow-deep-dive/
- https://ask.opendaylight.org/question/962/how-to-set-a-static-flow-in-karaf-and-how-to-get-information-about-configurated-flows/