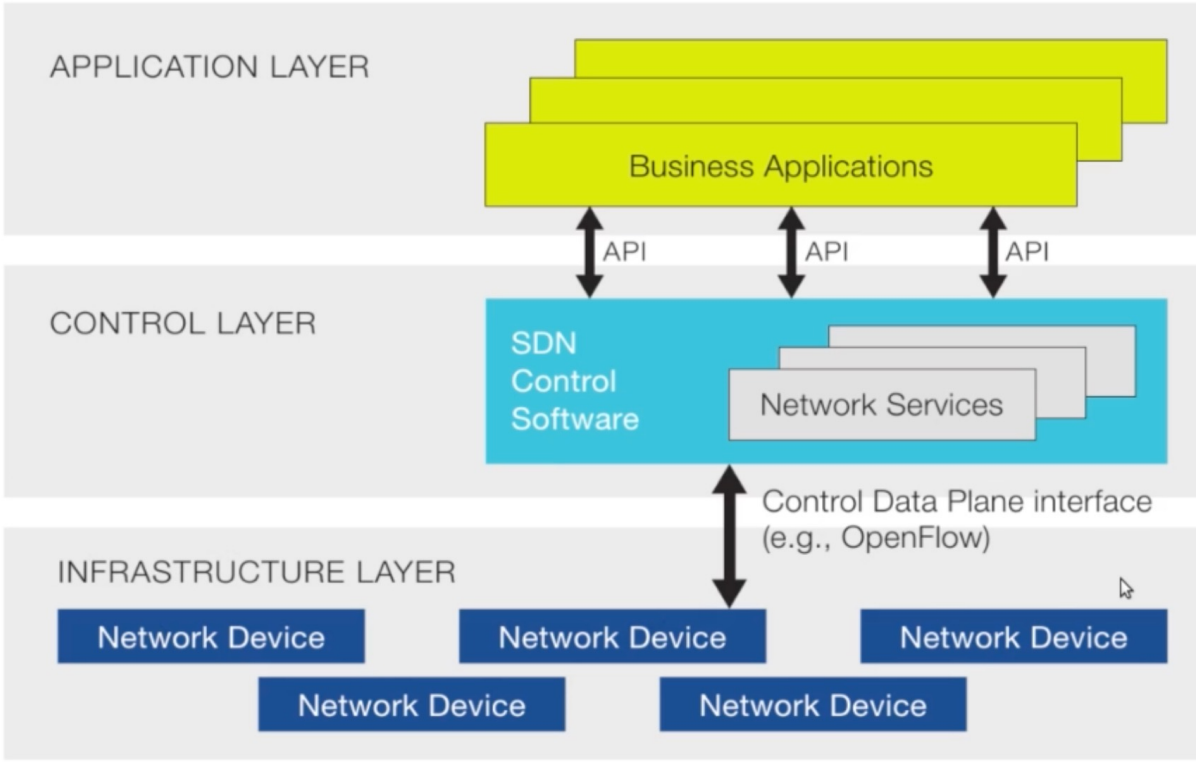
Software Defined Network SDN

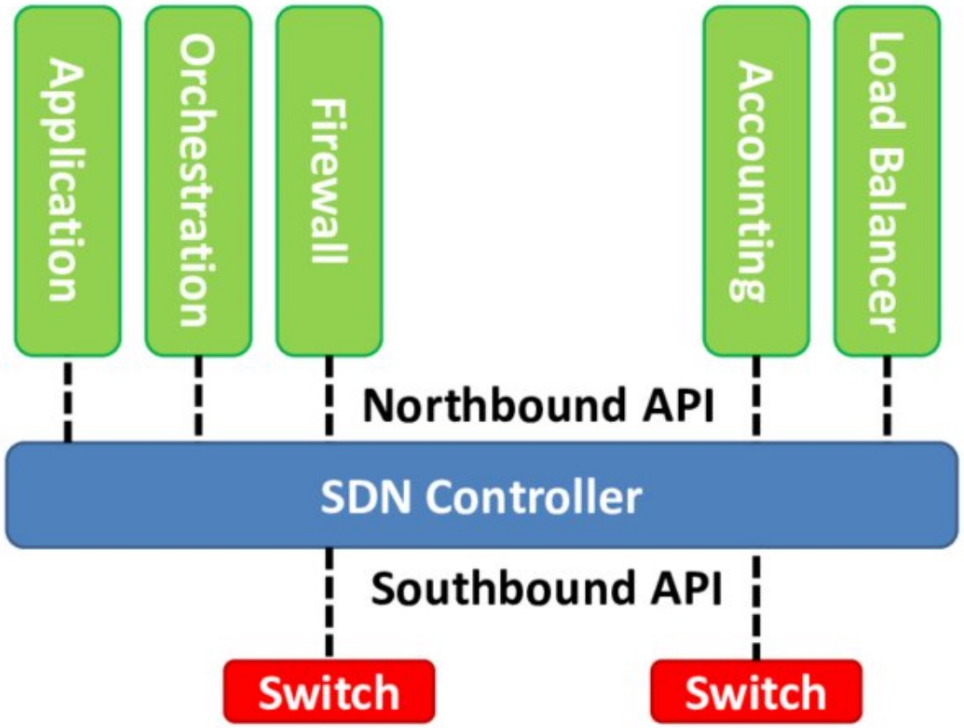
Ch 8

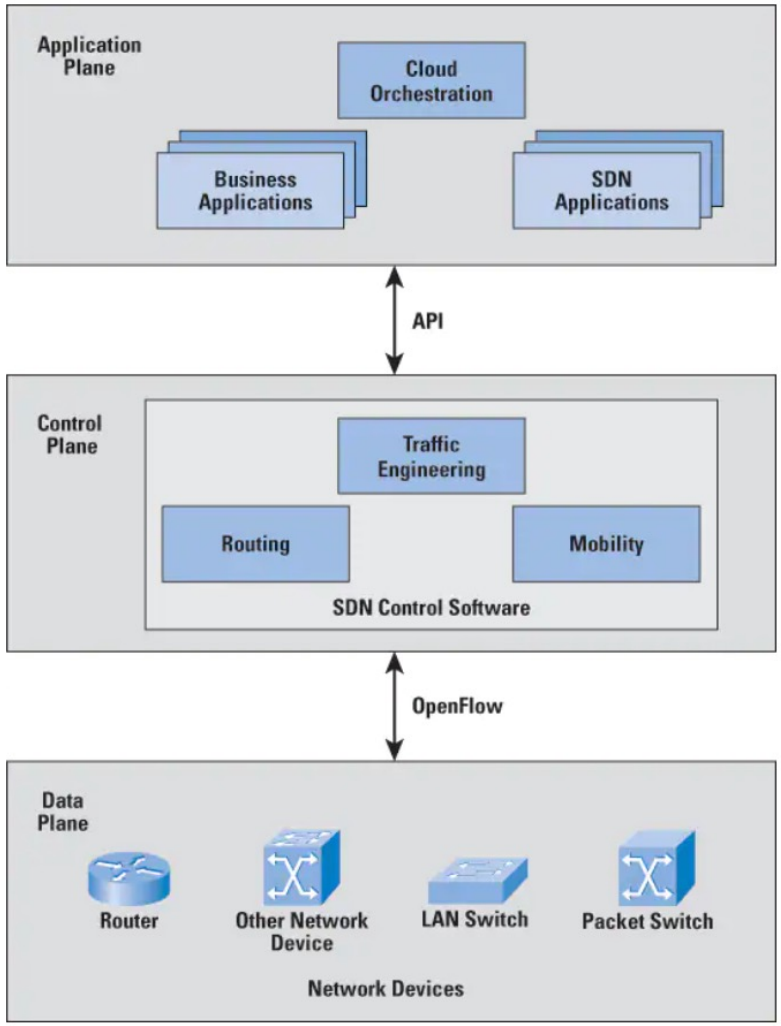
OpenDaylight Part I: Install OpenDaylight, Basic Command and OpenDaylight Plugin

SDN Architecture

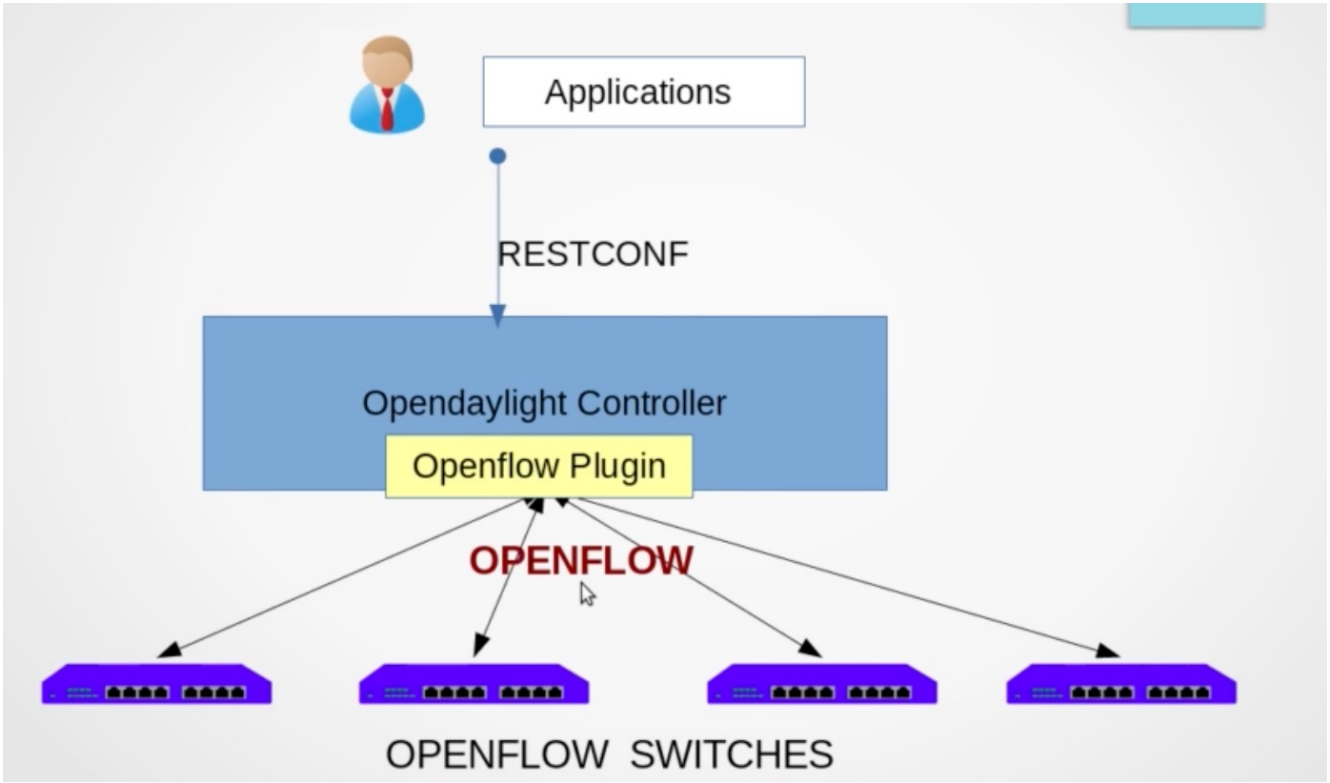


SDN Architecture

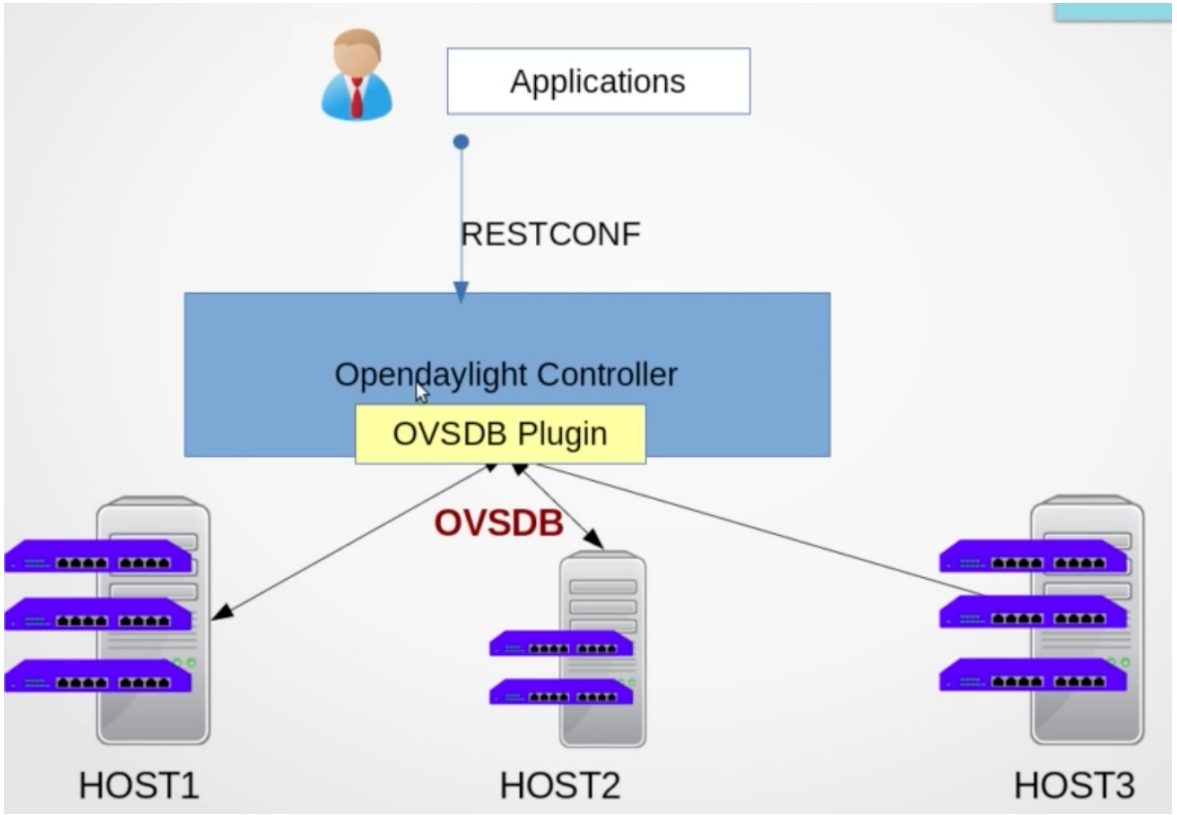




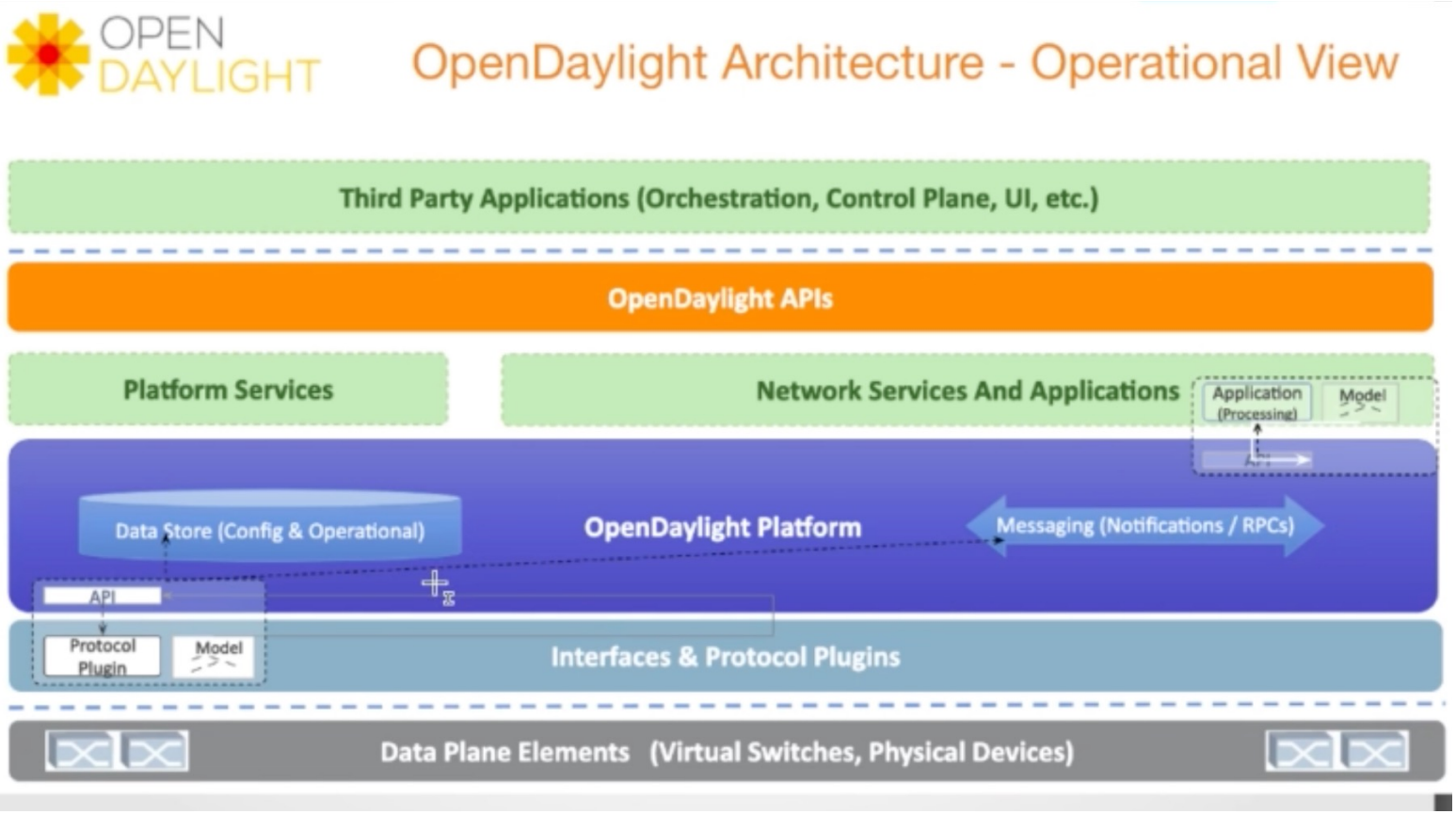
OpenDaylight Concept



OpenDaylight Concept



OpenDaylight Architecture



Installation

Preferred System requirements:

Configuration: 2 Core + 8GB RAM (preferably)

OS:

prefer Linux Ubuntu Version.

Ubuntu versions : Ubuntu 18.04
Java - 8+ version

JAVA Installation

```
sudo apt-get install default-jdk
```

Link เกี่ยวกับ OpenDaylight: https://docs.opendaylight.org/en/latest/index.html#

Installation

- ODL Installation
- OpenDaylight Version
  - wget https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integration/opendaylight/14.4.0/opendaylight-14.4.0.tar.gz
- tar xvzf opendaylight-14.4.0.tar.gz
- cd opendaylight-14.4.0/
- ./bin/karaf

You will see the karaf shell

Link ที่เกี่ยวข้อง: https://docs.opendaylight.org/en/latest/downloads.html

Console Commands

Feature command

Manage the features/plugins

```
feature:list
```

Lists all existing features available from the defined repositories.

```
feature:list -i
```

List only installed features

```
feature:list -i | grep name
```

List the installed features and grep with specific name.

Console Commands

Example

```
feature:list -i | grep restconf
```

> feature:install name

Installs a feature with the specified name and version.

Example

```
feature:install odl-restconf
```

Console Commands

> feature:uninstall name

Uninstalls a feature with the specified name and version

```
feature:uninstall odl-restconf

feature:info odl-restconf
```

Console Commands

Log command

log:display

Displays log entries.

log:get

Shows the currently set log level

log:set

Sets the log level.

The log level to set (TRACE, DEBUG, INFO, WARN, ERROR) or DEFAULT to unset

Log file is located in data/log/karaf.log

Exit Command

shutdown, logout

OpenFlow plugin Project

- Link ที่เกี่ยวข้อง https://docs.opendaylight.org/en/stable-neon/user-guide/openflow-plugin-project-user-guide.html

RestConf Methods

Standard mechanisms to allow Web applications to access the configuration data, state data, data-model-specific Remote Procedure Call (RPC) operations, and event notifications within a networking device, in a modular and extensible manner.

RESTCONF uses HTTP methods to provide CRUD operations on a conceptual datastore containing YANG-defined data, which is compatible with a server that implements NETCONF datastores.

Methods:

- Get
- Put
- Delete
- Post

REST Clients

1. POSTMAN

GUI Based.

To install, follow this link

- https://www.postman.com/downloads/
- https://linuxize.com/post/how-to-install-postman-on-ubuntu-18-04/

2. curl Quick to use. Command line interface. Little hard to understand the flags/options

```
sudo apt-get install curl
```

ODL Setup for demo

Plugin Installation

Install ODL as per previous session, and start the ODL KARAF shell

> ./bin/karaf

Install the restconf and Openflow plugins

```
feature:install odl-restconf
feature:install odl-openflowplugin-flow-services-rest
```

Default ODL username/password : **admin/admin**

ODL IP&Port: **http://192.168.122.219:8181**

GET http://192.168.122.219:8181restconf/operational/network-topology:network-topology/

DataModel: XML

Ip Address ที่ใช้ ให้เปลี่ยนเป็น IP Address ของเครื่องนักศึกษา เช่น http://172.16.161.137:8181

Accessing NorthBound RESTCONF/RESTAPI with POSTMAN

1. URL

```
http://192.168.122.227:8181/restconf/operational/network-topology:network-topology/
```

2. Base Headers:

In the Authorization TAB, Select,

```
Type: Basic Auth

Fill username and password as mentioned above
```

and click Preview Request,

3. In the Header TAB,

you could see "Authorization" Header and value "Basic xxxxxxxxxxxx" could have been populated automatically.

Add "Content-Type" and "Accept" header with Value is "application/xml" for both.

4. Method: GET and Query

Accessing NorthBound RESTCONF/RESTAPI with CURL

GET Method:

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET
http://192.168.122.227:8181/restconf/operational/network-topology:network-topology/
```

OpenFlow Plugin

Mininet

Installation

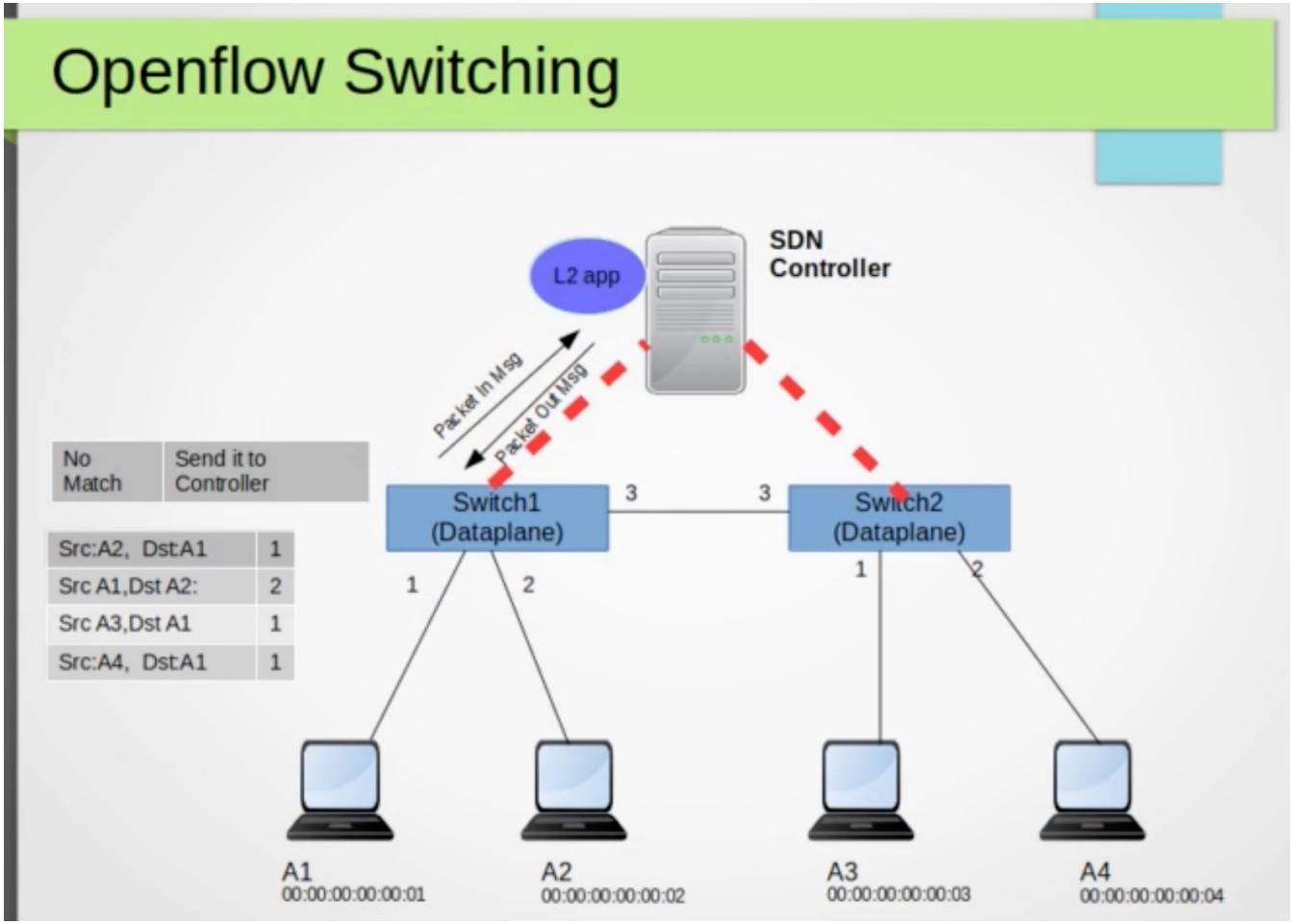
Open the terminal, and execute the below commands,

```
sudo apt-get update
sudo apt-get install -y openvswitch-switch
sudo apt-get install -y mininet
sudo apt-get install libxml2-utils
```

verify the installation

```
sudo mn --version
sudo ovs-vsctl show
```

OpenFlow Plugin



OpenFlow Plugin

OpenDaylight

Install the Openflow Plugin:

> *feature:install odl-mdsal-apidocs*
>
> *feature:install odl-restconf*
>
> *feature:install odl-openflowplugin-flow-services-rest*

optional plugins:

```
feature:install odl-openflowplugin-app-table-miss-enforcer
feature:install odl-openflowplugin-app-topology
feature:install odl-openflowplugin-app-topology-manager
feature:install odl-openflowplugin-app-lldp-speaker
feature:install odl-openflowplugin-app-topology-lldp-discovery
```

Verify openflow plugin is installed

In the karaf shell,

```
feature:list -i |grep openflowplugin
```

And in the terminal, verify the OPENFLOW port is listening mode,

```
sudo netstat -a | grep 6653
```

Quick verification

a) Run Mininet Single Topology

```
sudo mn --controller=remote,ip=127.0.0.1 --mac -i 10.1.1.0/24 --switch=ovsk,protocols=OpenFlow13 --
topo=single,4
```

b) Verify the Switch - Controller connection status

```
sudo ovs-vsctl show
```

Quick verification

c) verify the Openflow Northbound API - INVENTORY Endpoint , using curl or apidoc

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/ | xmllint --format  -
```

d) verify the Openflow Northbound API - Topology Endpoint , using curl or apidoc

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET
http://localhost:8181/restconf/operational/network-topology:network-topology/  | xmllint --format  -
```

REST API - apidocs

http://192.168.122.227:8181/apidoc/explorer/index.html

username: admin password: admin

Verirfy Topology endpoint

/restconf/operational/network-topology:network-topology/

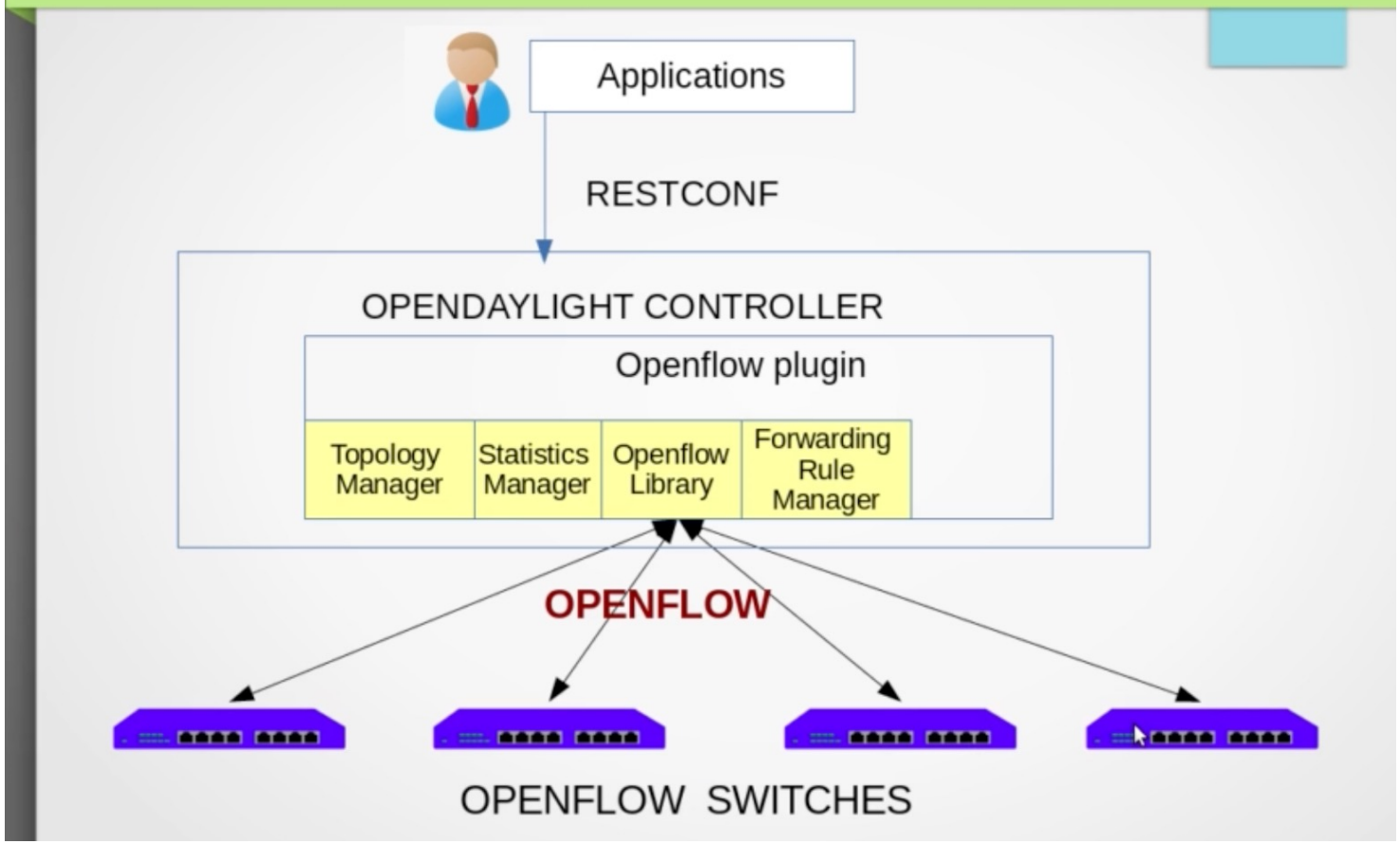
Verirfy Inventory endpoint

/restconf/operational/opendaylight-inventory:nodes/

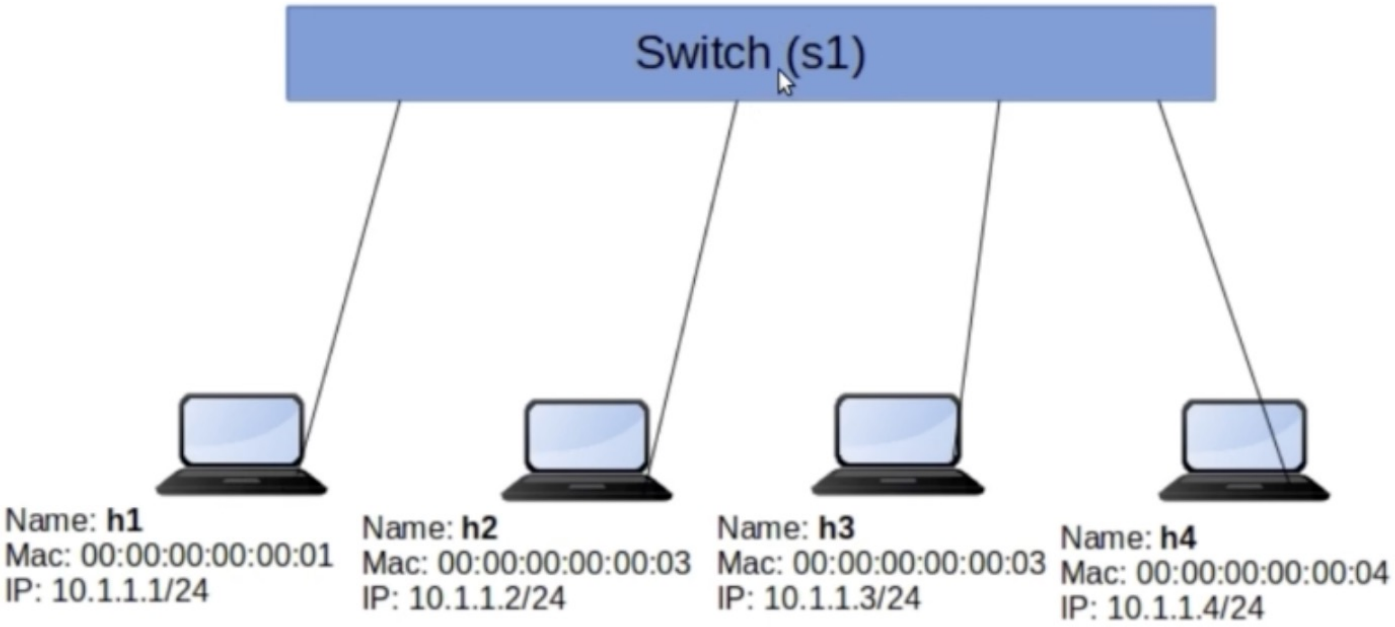
Terminologies

- Topology Discovery
  - Discovers the switches, Ports, and Links
- Statistics
  - Collect the Switch/Port Statistics , Tables(Flow, group, meter etc) Statistics etc.
- OpenFlows/Rules programming
  - Install/Modify/Delete the Openflow Flows/Group/meter etc

Architecture



Topology discovery

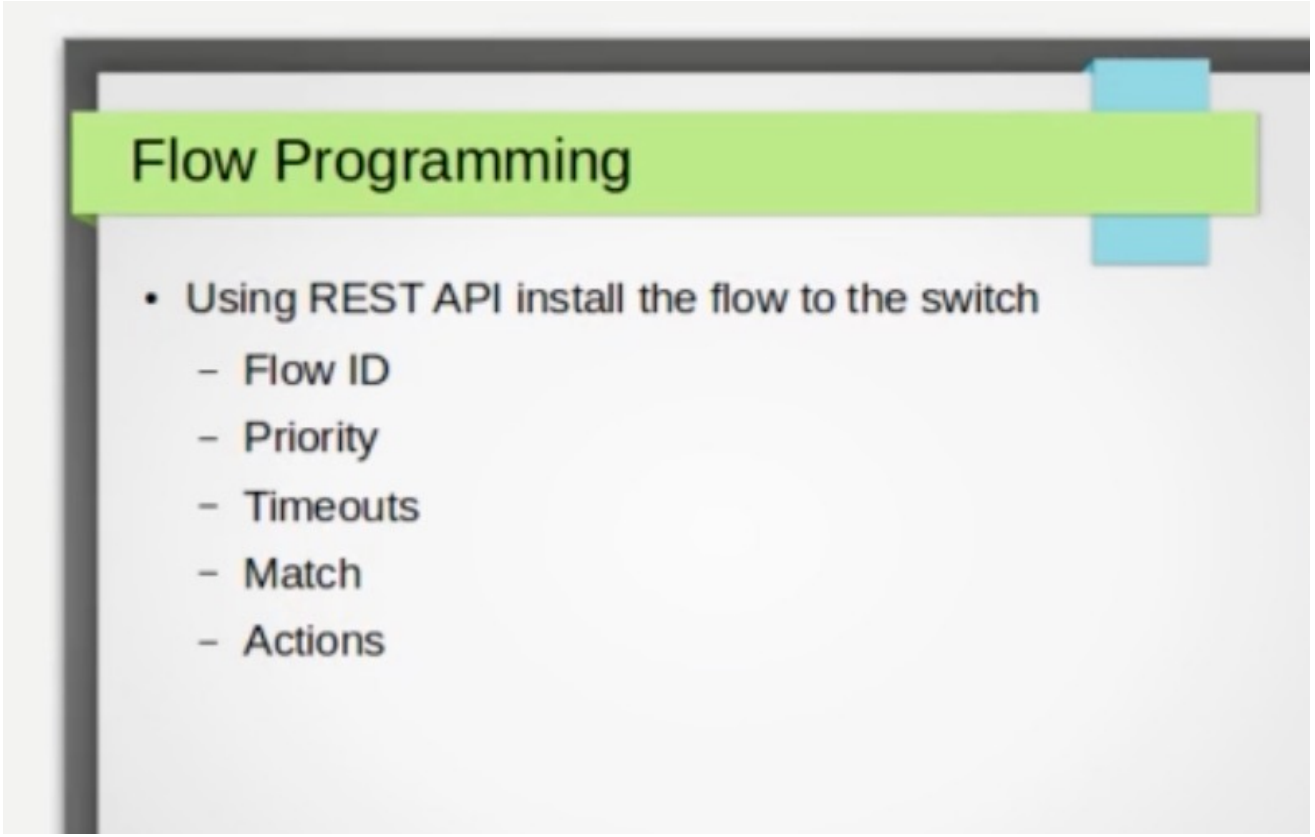


Topology Discovery

- Topology name
- Links
  - source : switch 1, port 2 , destination : switch 2 , port 1
- Switches
  - Ports
    - Port1
    - port2

Statistics

- Switch Details/Description
- Port Statistics
- Tables/Flows statistics


Flow Programming
• Using REST API install the flow to the switch
– Flow ID
– Priority
– Timeouts
– Match
– Actions

Architecture

- **Forwarding Rules Manager:**
  - Main entity that manages the OpenFlow switch inventory and the configuration (programming) of flows in switches.
  - It also reconciles user configuration with network state discovered by the OpenFlow plugin.

OpenDaylight OpenFlow

- Lab Example includes as follows:
  - Hub
  - L2 Switch
  - L3 Switch
  - L4 Switch

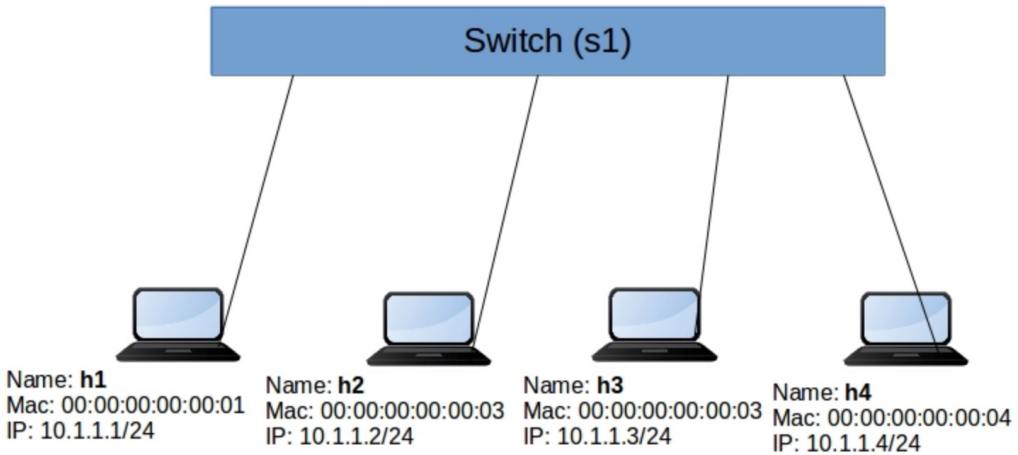
Hub Exercise

1. Objective

Perform FLOOD action in the switch.

2. Create a Single topology

```
sudo mn --controller=remote,ip=127.0.0.1 --mac -i 10.1.1.0/24 --switch=ovsk,protocols=OpenFlow13 --
topo=single,4
```



Hub Exercise

3. Make sure ODL is running and openflow plugin is installed

In the karaf shell,

```
feature:list -i |grep odl-openflowplugin-flow-services-rest
```

And in the terminal, verify the OPENFLOW port is listening mode,

```
sudo netstat -a | grep 6653
```

4. Check the switches are connected to the ODL Controller

```
sudo ovs-vsctl show
```

Hub Exercise

5. Add a flood flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1 -d
@hub.xml
```

6. Verify the flows in OVS

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

7. Try pingall from mininet shell

Hub Exercise

8. verify the INVENTORY Endpoint , using curl or apidoc

Config datastore

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/ | xmllint --format  -
```

Operation Datastore

All nodes:

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET
http://localhost:8181/restconf/operational/opendaylight-inventory:nodes/ | xmllint --format  -
```

Specific flow:

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET
http://localhost:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1
| xmllint --format  -
```

Hub Exercise

9. verify the Topology Endpoint , using curl or apidoc

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET
http://localhost:8181/restconf/operational/network-topology:network-topology/  | xmllint --format  -
```

10. Remove the flood flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X DELETE
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1
```

11. exit from mininet

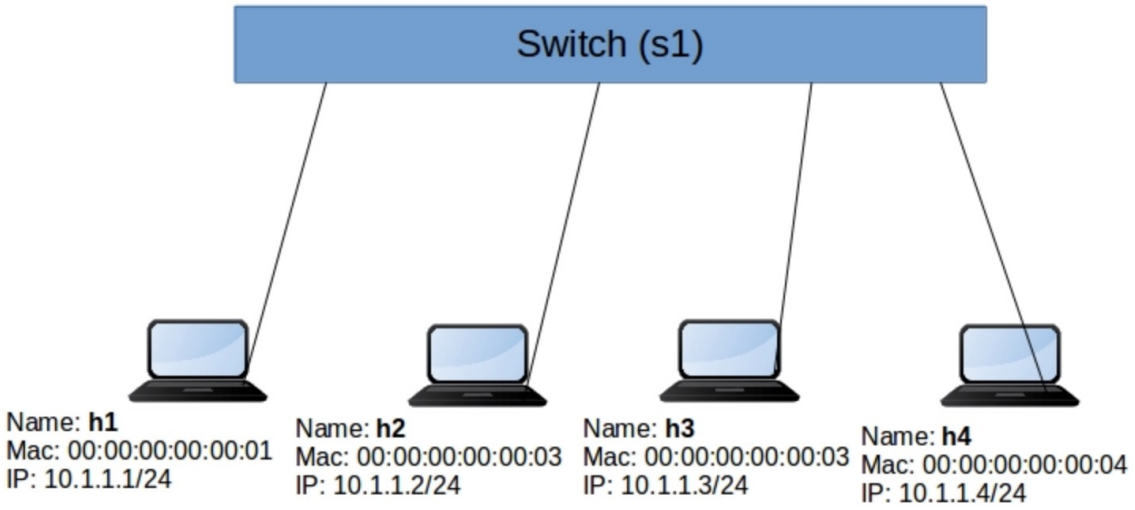
```
exit
```

12. Verify exit the topology(without removing the flow from ODL) and test again.

L2 Switch Exercise

1. Create a Single topology

```
sudo mn --controller=remote,ip=127.0.0.1 --mac -i 10.1.1.0/24 --switch=ovsk,protocols=OpenFlow13 --
topo=single,4
```



L2 Switch Exercise

2. Make sure ODL is running and openflow plugin is installed

In the karaf shell,

```
feature:list -i |grep openflow
```

And in the terminal, verify the OPENFLOW port is listening mode,

```
sudo netstat -i | grep 6653
```

3. Check the switches are connected to the ODL Controller

```
sudo ovs-vsctl show
```

L2 Switch Exercise

4. Add a ARP flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1 -d
@arp.xml
```

5. Add a h1(00:00:00:00:00:01) to h2(00:00:00:00:00:02) flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/2 -d
@l2match1.xml
```

6. Add a h2(00:00:00:00:00:02) to h1(00:00:00:00:00:01) flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/3 -d
@l2match2.xml
```

L2 Switch Exercise

7. Verify the flows in OVS

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

8. Verify the flows in ODL Datastore

Specific flow:

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml"
-X GET  http://localhost:8181/restconf/operational/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1 | xmllint --format  -
```

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml"
-X GET  http://localhost:8181/restconf/operational/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/2 | xmllint --format  -
```

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml"
-X GET  http://localhost:8181/restconf/operational/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/3 | xmllint --format  -
```

L2 Switch Exercise

9. Try ping from mininet shell

```
h1 ping h2
```

10. Delete the flows from ODL

```
curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1

curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/2

curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/3
```

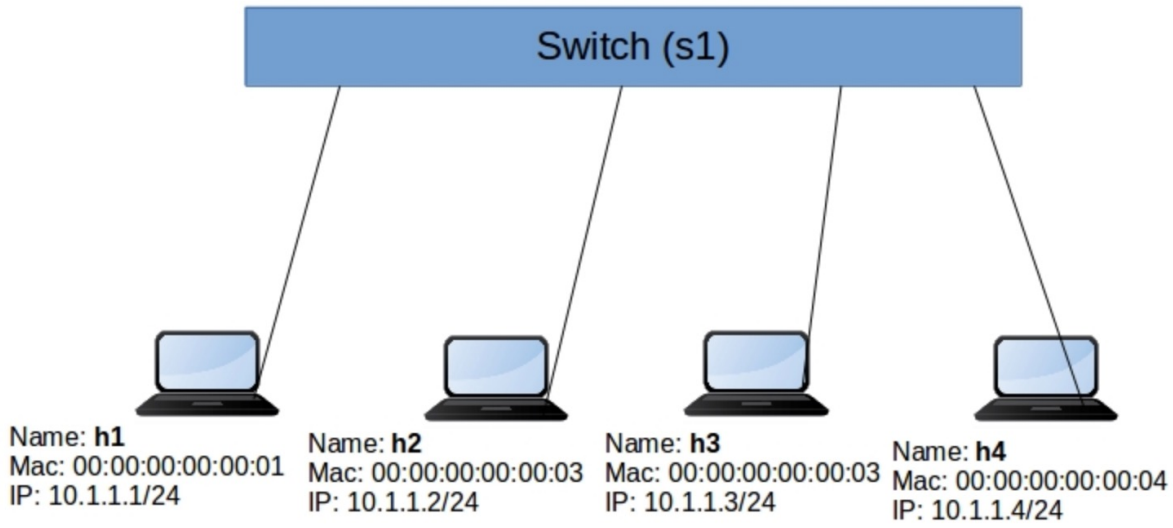
11. exit from mininet

```
exit
```

L3 Switch Exercise

1. Create a Single topology

```
sudo mn --controller=remote,ip=127.0.0.1 --mac -i 10.1.1.0/24 --switch=ovsk,protocols=OpenFlow13 --
topo=single,4
```



L3 Switch Exercise

2. Make sure ODL is running and openflow plugin is installed

In the karaf shell,

```
feature:list -i |grep openflow
```

And in the terminal, verify the OPENFLOW port is listening mode,

```
sudo netstat -i | grep 6653
```

3. Check the switches are connected to the ODL Controller

```
sudo ovs-vsctl show
```

L3 Switch Exercise

4. Add a ARP flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1 -d
@arp.xml
```

5. Add a h1(10.1.1.1) to h2(10.1.1.2) flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/2 -d
@l3match1.xml
```

6. Add a h2(10.1.1.2) to h1(10.1.1.1) flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/3 -d
@l3match2.xml
```

L3 Switch Exercise

7. Verify the flows in OVS

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

8. Try ping from mininet shell

```
h1 ping h2
```

9. Delete the flows from ODL

```
curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1

curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/2

curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/3
```

10. exit from mininet

```
exit
```

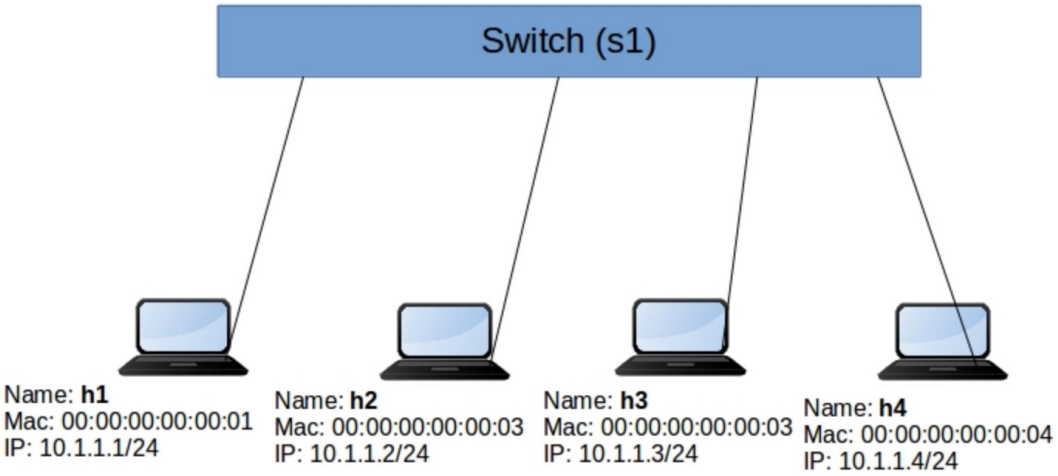
L4 Switch Exercise

1. Objective

Allow only TCP Traffic between h1 and h2

2. Create a Single topology

```
sudo mn --controller=remote,ip=127.0.0.1 --mac -i 10.1.1.0/24 --switch=ovsk,protocols=OpenFlow13 --
topo=single,4
```



L4 Switch Exercise

3. Make sure ODL is running and openflow plugin is installed

In the karaf shell,

```
feature:list -i |grep openflow
```

And in the terminal, verify the OPENFLOW port is listening mode,

```
sudo netstat -i | grep 6653
```

4. Check the switches are connected to the ODL Controller

```
sudo ovs-vsctl show
```

L4 Switch Exercise

5. Add a ARP flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1 -d
@arp.xml
```

6. Add a h1(10.1.1.1) to h2(10.1.1.2) TCP flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/2 -d
@l4match1.xml
```

7. Add a h2(10.1.1.2) to h1(10.1.1.1) TCP flow

```
curl --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X PUT
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/3 -d
@l4match2.xml
```

L4 Switch Exercise

8. Verify the flows in OVS

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

9. Try TCP traffic in mininet shell

```
h1 iperf -s &
h2 iperf -c h1
```

L4 Switch Exercise

10. Delete the flows from ODL

```
curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1

curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/2

curl -v --user "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X
DELETE  http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/3
```

11. exit from mininet

```
exit
```