

Analysing L3 routing on a SDN environment

Lesson 4

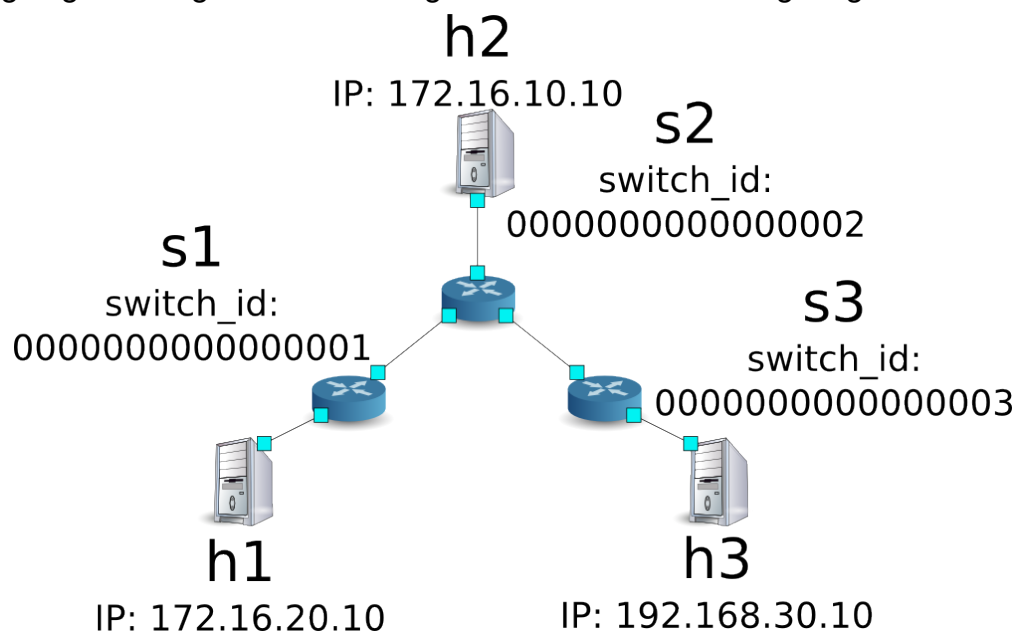
Please don't write anything on this paper!

Objectives:

The objective of this task to understand basic static routing on a L3 Routed network. We will learn how to use the REST API to control the routing between multiple broadcast domains

Example of the Operation of a Single Tenant network

We are going to configure the following network on the following diagram:



Building the environment:

First, build an environment on Mininet. Parameters of the mn command are as follows.

Parameter	Value	Explanation
topo	linear,3	Topology where three switches are connected in serial
mac	None	Set the MAC address of the host automatically
switch	ovsk	Use Open vSwitch + Set protocol to OpenFlow 13
controller	remote	Use an external one for OpenFlow controller
x	None	Start xterm

An execution example is as follows.

```
ryu@ryu-vm:~$ sudo mn --topo linear,3 --mac --switch ovsk,protocols=OpenFlow13 --controller remote -x
```

See the address scheme lower in this document.

Delete the IP address that is assigned automatically on each host and set a new IP address.

On host: h1:

```
root@ryu-vm:~# ip addr del 10.0.0.1/8 dev h1-eth0
root@ryu-vm:~# ip addr add 172.16.20.10/24 dev h1-eth0
```

On host: h2:

```
root@ryu-vm:~# ip addr del 10.0.0.2/8 dev h2-eth0
root@ryu-vm:~# ip addr add 172.16.10.10/24 dev h2-eth0
```

On host: h3:

```
root@ryu-vm:~# ip addr del 10.0.0.3/8 dev h3-eth0
root@ryu-vm:~# ip addr add 192.168.30.10/24 dev h3-eth0
```

Start the REST Router application:

controller: c0 (root):

```
root@ryu-vm:~# ryu-manager ryu.app.rest_router
```

Refer to the output log and examine the flow tables on all routers.

Setting the Address

Set the address for each router.

First, set the addresses “172.16.20.1/24” and “172.16.30.30/24” for router s1

Send a POST request to **[controller]/router/[bridge_id]**

With data:

```
{
    "address": "[ip_address]/[network_mask]"
}
```

Node: c0 (root):

```
root@ryu-vm:~# curl -X POST -d '{"address": "172.16.20.1/24"}'
http://localhost:8080/router/000000000000000001
```

```
root@ryu-vm:~# curl -X POST -d '{"address": "172.16.30.30/24"}'
http://localhost:8080/router/000000000000000001
```

Next, set the addresses “172.16.10.1/24”, “172.16.30.1/24” and “192.168.10.1/24” for router s2.

```
root@ryu-vm:~# curl -X POST -d '{"address": "172.16.10.1/24"}'  
http://localhost:8080/router/000000000000000002
```

```
root@ryu-vm:~# curl -X POST -d '{"address": "172.16.30.1/24"}'  
http://localhost:8080/router/000000000000000002
```

```
root@ryu-vm:~# curl -X POST -d '{"address": "192.168.10.1/24"}'  
http://localhost:8080/router/000000000000000002
```

Then, set the addresses “192.168.30.1/24” and “192.168.10.20/24” for router s3.

Node: c0 (root):

```
root@ryu-vm:~# curl -X POST -d '{"address": "192.168.30.1/24"}'  
http://localhost:8080/router/000000000000000003
```

```
root@ryu-vm:~# curl -X POST -d '{"address": "192.168.10.20/24"}'  
http://localhost:8080/router/000000000000000003
```

IP addresses to the router have been set. Register as the default gateway on each host.

host: h1:

```
root@ryu-vm:~# ip route add default via 172.16.20.1
```

host: h2:

```
root@ryu-vm:~# ip route add default via 172.16.10.1
```

host: h3:

```
root@ryu-vm:~# ip route add default via 192.168.30.1
```

Configuring the Default Route

Set the default route for edge routers.

Send a POST request to **[controller]/router/[bridge_id]**

With data:

```
{  
    "gateway": "[ip_address]"  
}
```

First, set router s2 as the default route of router s1.

Node: c0 (root):

```
root@ryu-vm:~# curl -X POST -d '{"gateway": "172.16.30.1"}'  
http://localhost:8080/router/000000000000000001
```

Set router s2 as the default route of router s3.

Node: c0 (root):

```
root@ryu-vm:~# curl -X POST -d '{"gateway": "192.168.10.1"}'  
http://localhost:8080/router/000000000000000003
```

Setting Static Routes

For s2 router, set a static route to the network (192.168.30.0/24) under router s3 and network (172.16.20.0/24) under s1

Send a POST request to **[controller]/router/[bridge_id]**

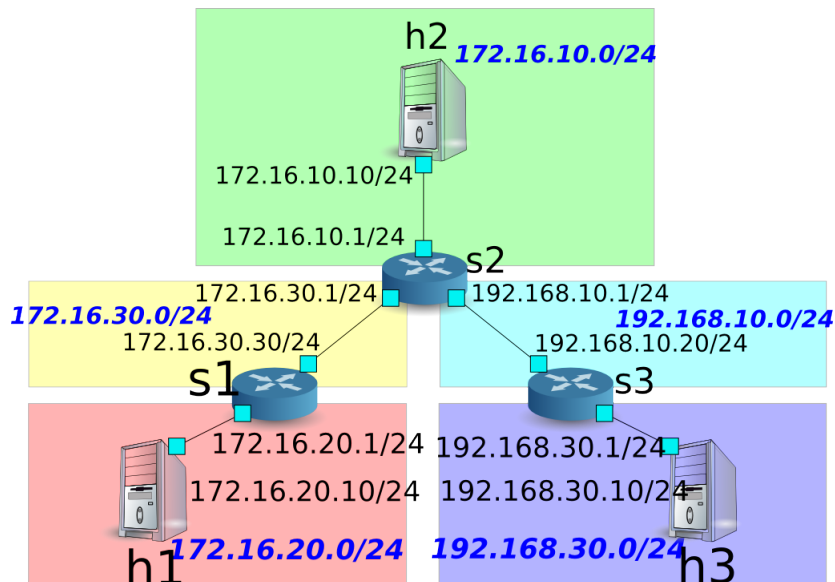
With data:

```
{  
    "destination": "[network]/[mask]",  
    "gateway": "[next_hop_ip]"  
}
```

Node: c0 (root):

```
root@ryu-vm:~# curl -X POST -d '{"destination": "192.168.30.0/24", "gateway":  
"192.168.10.20"}' http://localhost:8080/router/000000000000000002  
root@ryu-vm:~# curl -X POST -d '{"destination": "172.16.20.0/24", "gateway":  
"172.16.30.30"}' http://localhost:8080/router/000000000000000002
```

Your address scheme over the topology should be the following:



Verifying the Setting

Verify the results by pinging between each node.

Eg h1 ping h2, etc

There might be an unidentified issue with pingall. Separate ping commands should work

Try to ping some network interfaces on the forwarders too and check what is happening on the forwarders. (Refer to console log on the controller)

3 points