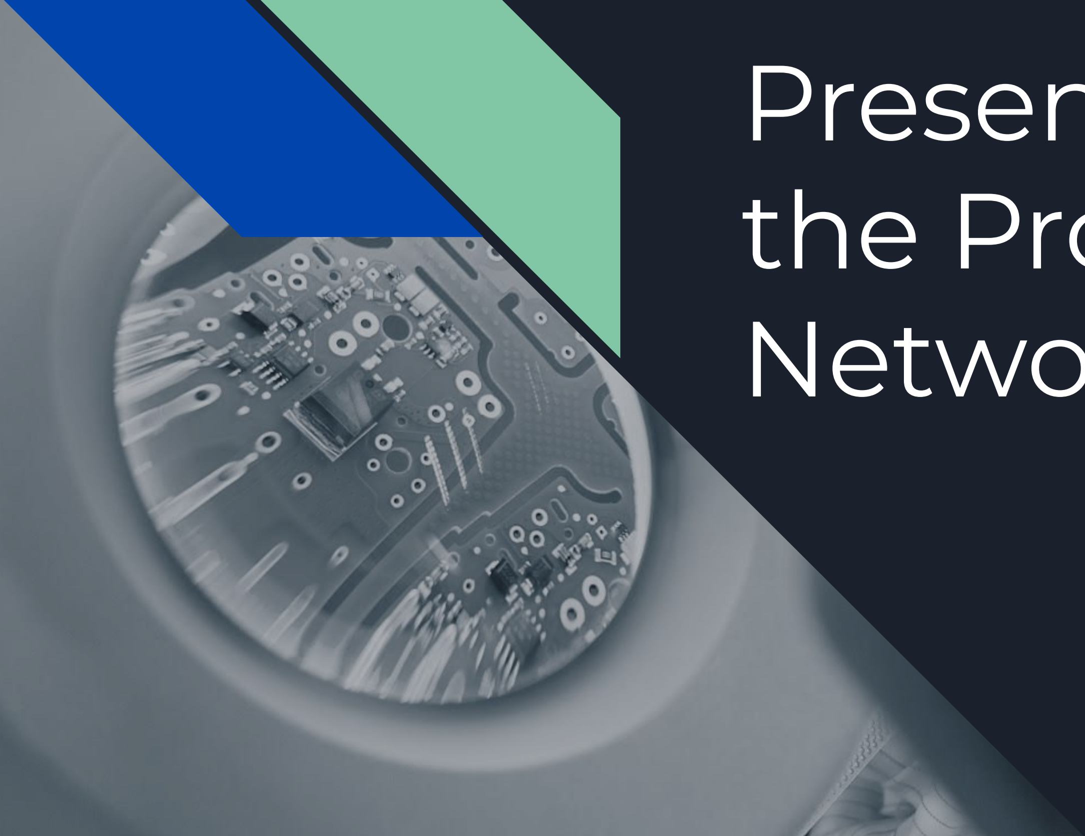


Presentation of the Project for Networking 2

Aloisi Deborah
Dalla Palma Mathias

SUMMARY

Introduction

RYU API Solution

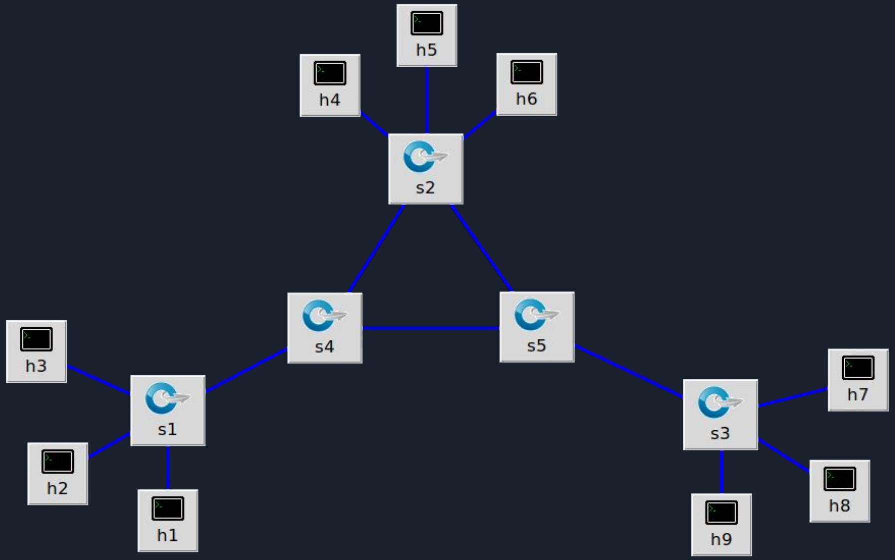
Morphing Network

Introduction

- To enable RYU SDN controller to build network and modify its topology
- To enable RYU SDN controller to build network slices
- To consider that each network node might host «services», that in this case will be represented by virtual switches/routers
- The SDN controller will virtually re-program connectivity of the physical network

RYU API Solution

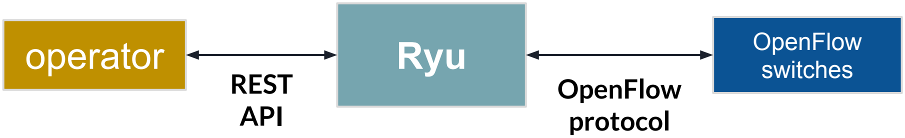
- Mininet Network



❖ Hosts → 9

❖ Switches → 5

Restful API Used



operator

REST API

Ryu

OpenFlow protocol

OpenFlow switches

```
$  curl -X DELETE http://localhost:8080/stats/flowentry/clear/<dpid>
```

Delete all flow entries of the switch which specified with Datapath ID in URI.

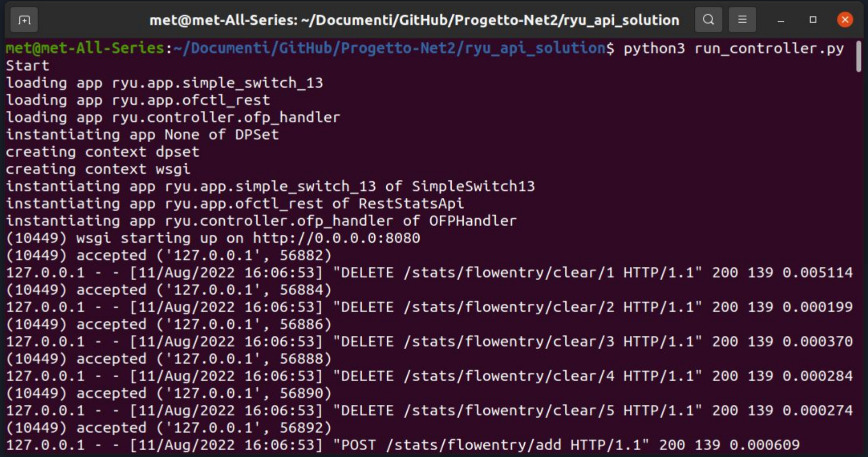
```
 $ curl -X POST -d '{
    "dpid": 1,
    "match":{
       "dl_dst": "00:00:00:00:00:01"
    },
    "actions":[
       {
          "type":"OUTPUT",
             "port": 2
       }
   ]
 }' http://localhost:8080/stats/flowentry/add
```

Add a flow entry to the switch.

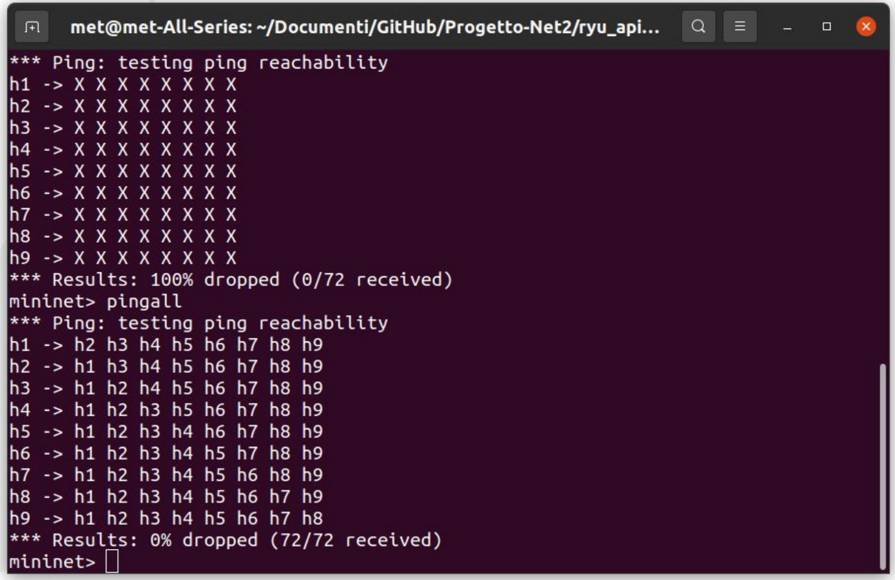
Run the net

$ sudo mn --custom ryu_api_solution/network.py --controller remote --topo test --arp

$ python3 ryu_api_solution/run_controller.py


met@met-All-Series: ~/Documenti/GitHub/Progetto-Net2/ryu_api_solution

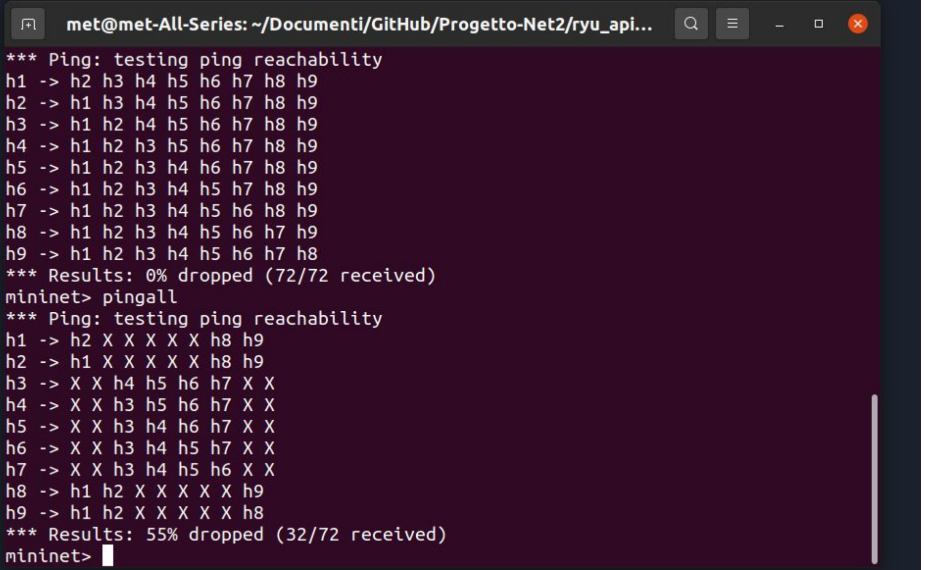
```
met@met-All-Series:~/Documenti/GitHub/Progetto-Net2/ryu_api_solution$ python3 run_controller.py
Start
loading app ryu.app.simple_switch_13
loading app ryu.app.ofctl_rest
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
creating context wsgi
instantiating app ryu.app.simple_switch_13 of SimpleSwitch13
instantiating app ryu.app.ofctl_rest of RestStatsApi
instantiating app ryu.controller.ofp_handler of OFPHandler
(10449) wsgi starting up on http://0.0.0.0:8080
(10449) accepted ('127.0.0.1', 56882)
127.0.0.1 - - [11/Aug/2022 16:06:53] "DELETE /stats/flowentry/clear/1 HTTP/1.1" 200 139 0.005114
(10449) accepted ('127.0.0.1', 56884)
127.0.0.1 - - [11/Aug/2022 16:06:53] "DELETE /stats/flowentry/clear/2 HTTP/1.1" 200 139 0.000199
(10449) accepted ('127.0.0.1', 56886)
127.0.0.1 - - [11/Aug/2022 16:06:53] "DELETE /stats/flowentry/clear/3 HTTP/1.1" 200 139 0.000370
(10449) accepted ('127.0.0.1', 56888)
127.0.0.1 - - [11/Aug/2022 16:06:53] "DELETE /stats/flowentry/clear/4 HTTP/1.1" 200 139 0.000284
(10449) accepted ('127.0.0.1', 56890)
127.0.0.1 - - [11/Aug/2022 16:06:53] "DELETE /stats/flowentry/clear/5 HTTP/1.1" 200 139 0.000274
(10449) accepted ('127.0.0.1', 56892)
127.0.0.1 - - [11/Aug/2022 16:06:53] "POST /stats/flowentry/add HTTP/1.1" 200 139 0.000609
```


met@met-All-Series: ~/Documenti/GitHub/Progetto-Net2/ryu_api...

```
*** Ping: testing ping reachability
h1 -> X X X X X X X X
h2 -> X X X X X X X X
h3 -> X X X X X X X X
h4 -> X X X X X X X X
h5 -> X X X X X X X X
h6 -> X X X X X X X X
h7 -> X X X X X X X X
h8 -> X X X X X X X X
h9 -> X X X X X X X X
*** Results: 100% dropped (0/72 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9
h2 -> h1 h3 h4 h5 h6 h7 h8 h9
h3 -> h1 h2 h4 h5 h6 h7 h8 h9
h4 -> h1 h2 h3 h5 h6 h7 h8 h9
h5 -> h1 h2 h3 h4 h6 h7 h8 h9
h6 -> h1 h2 h3 h4 h5 h7 h8 h9
h7 -> h1 h2 h3 h4 h5 h6 h8 h9
h8 -> h1 h2 h3 h4 h5 h6 h7 h9
h9 -> h1 h2 h3 h4 h5 h6 h7 h8
*** Results: 0% dropped (72/72 received)
mininet> 
```

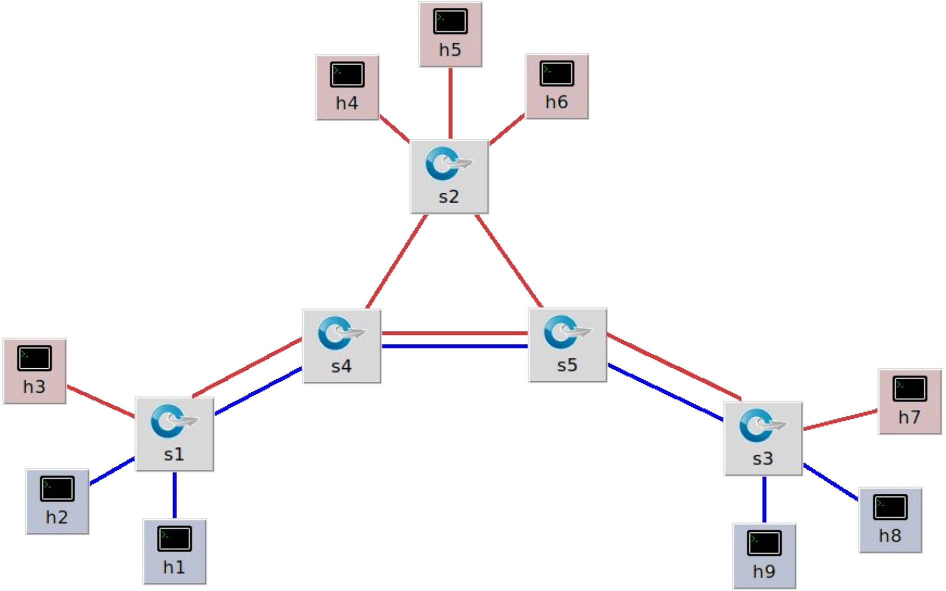
Slicing the net

$ python3 ryu_api_solution/slice_topology.py

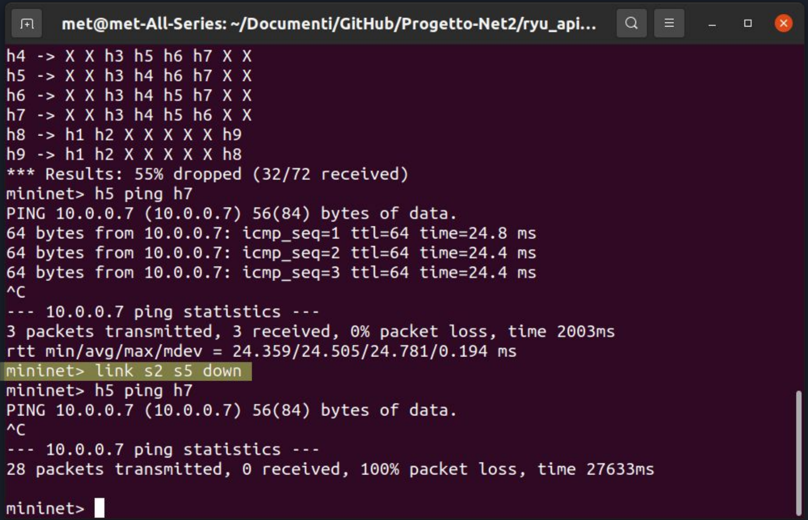


met@met-All-Series: ~/Documenti/GitHub/Progetto-Net2/ryu_api...

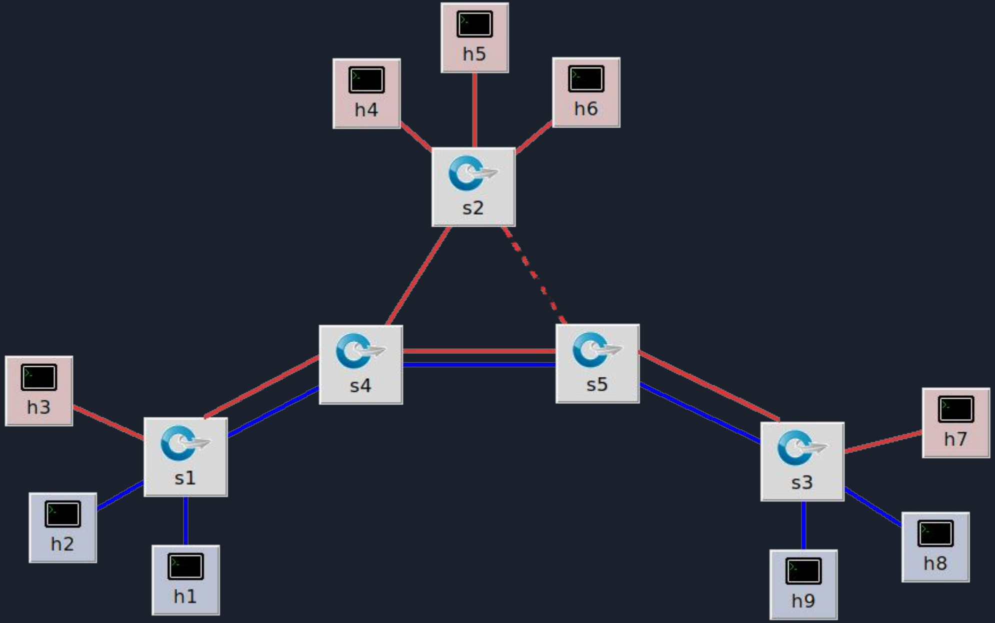
```
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9
h2 -> h1 h3 h4 h5 h6 h7 h8 h9
h3 -> h1 h2 h4 h5 h6 h7 h8 h9
h4 -> h1 h2 h3 h5 h6 h7 h8 h9
h5 -> h1 h2 h3 h4 h6 h7 h8 h9
h6 -> h1 h2 h3 h4 h5 h7 h8 h9
h7 -> h1 h2 h3 h4 h5 h6 h8 h9
h8 -> h1 h2 h3 h4 h5 h6 h7 h9
h9 -> h1 h2 h3 h4 h5 h6 h7 h8
*** Results: 0% dropped (72/72 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X X X X h8 h9
h2 -> h1 X X X X X h8 h9
h3 -> X X h4 h5 h6 h7 X X
h4 -> X X h3 h5 h6 h7 X X
h5 -> X X h3 h4 h6 h7 X X
h6 -> X X h3 h4 h5 h7 X X
h7 -> X X h3 h4 h5 h6 X X
h8 -> h1 h2 X X X X X h9
h9 -> h1 h2 X X X X X h8
*** Results: 55% dropped (32/72 received)
mininet> 
```



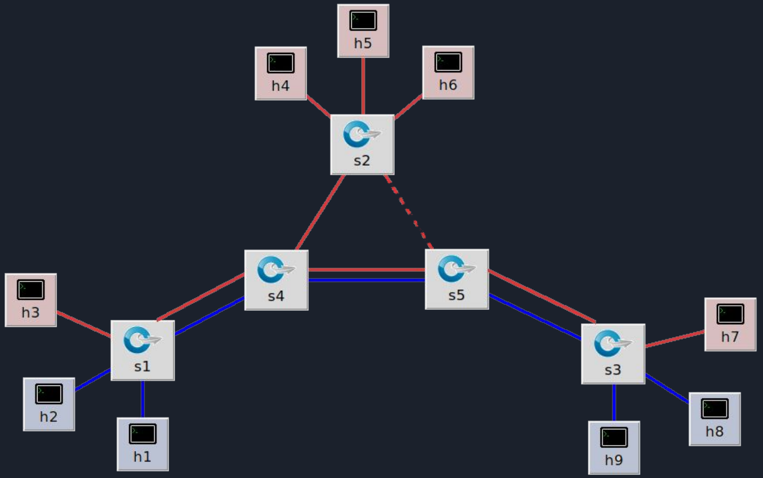
Simulating a link down


met@met-All-Series: ~/Documenti/GitHub/Progetto-Net2/ryu_api...

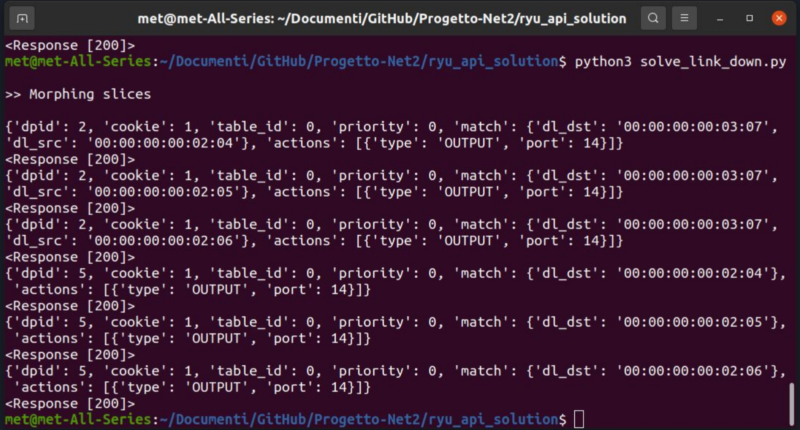
```
h4 -> X X h3 h5 h6 h7 X X
h5 -> X X h3 h4 h6 h7 X X
h6 -> X X h3 h4 h5 h7 X X
h7 -> X X h3 h4 h5 h6 X X
h8 -> h1 h2 X X X X X h9
h9 -> h1 h2 X X X X X h8
*** Results: 55% dropped (32/72 received)
mininet> h5 ping h7
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=1 ttl=64 time=24.8 ms
64 bytes from 10.0.0.7: icmp_seq=2 ttl=64 time=24.4 ms
64 bytes from 10.0.0.7: icmp_seq=3 ttl=64 time=24.4 ms
^C
--- 10.0.0.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 24.359/24.505/24.781/0.194 ms
mininet> link s2 s5 down
mininet> h5 ping h7
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
^C
--- 10.0.0.7 ping statistics ---
28 packets transmitted, 0 received, 100% packet loss, time 27633ms

mininet> 
```



Solve the link down

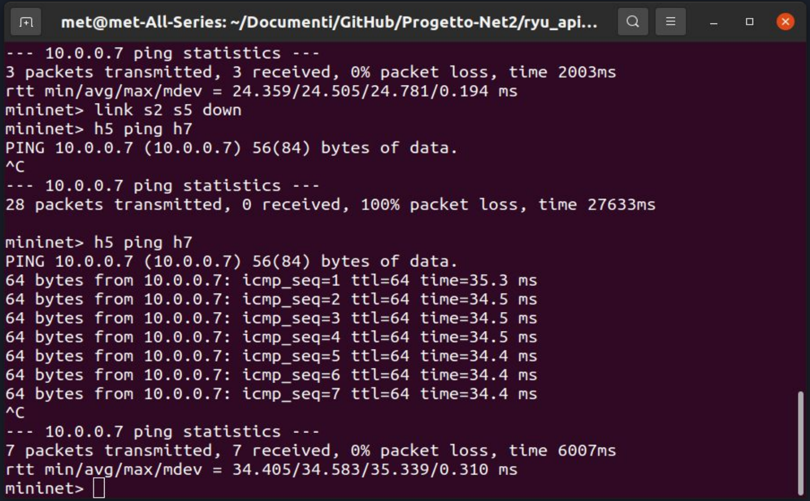


Host: $ python3 ryu_api_solution/solve_link_down.py



met@met-All-Series: ~/Documenti/GitHub/Progetto-Net2/ryu_api_solution

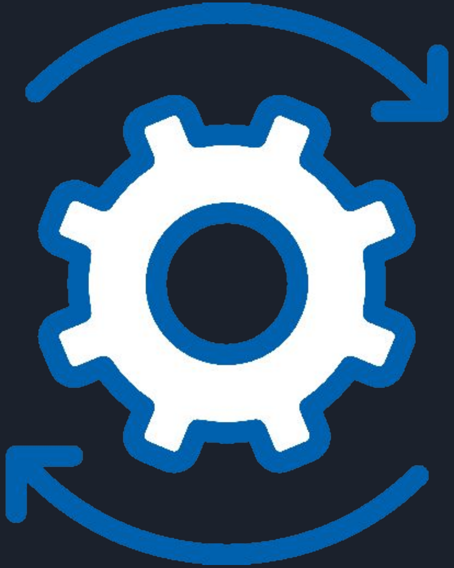
```
<Response [200]>
met@met-All-Series:~/Documenti/GitHub/Progetto-Net2/ryu_api_solution$ python3 solve_link_down.py

>> Morphing slices

{'dpid': 2, 'cookie': 1, 'table_id': 0, 'priority': 0, 'match': {'dl_dst': '00:00:00:00:03:07',
'dl_src': '00:00:00:00:02:04'}, 'actions': [{'type': 'OUTPUT', 'port': 14}]}
<Response [200]>
{'dpid': 2, 'cookie': 1, 'table_id': 0, 'priority': 0, 'match': {'dl_dst': '00:00:00:00:03:07',
'dl_src': '00:00:00:00:02:05'}, 'actions': [{'type': 'OUTPUT', 'port': 14}]}
<Response [200]>
{'dpid': 2, 'cookie': 1, 'table_id': 0, 'priority': 0, 'match': {'dl_dst': '00:00:00:00:03:07',
'dl_src': '00:00:00:00:02:06'}, 'actions': [{'type': 'OUTPUT', 'port': 14}]}
<Response [200]>
{'dpid': 5, 'cookie': 1, 'table_id': 0, 'priority': 0, 'match': {'dl_dst': '00:00:00:00:02:04'},
 'actions': [{'type': 'OUTPUT', 'port': 14}]}
<Response [200]>
{'dpid': 5, 'cookie': 1, 'table_id': 0, 'priority': 0, 'match': {'dl_dst': '00:00:00:00:02:05'},
 'actions': [{'type': 'OUTPUT', 'port': 14}]}
<Response [200]>
{'dpid': 5, 'cookie': 1, 'table_id': 0, 'priority': 0, 'match': {'dl_dst': '00:00:00:00:02:06'},
 'actions': [{'type': 'OUTPUT', 'port': 14}]}
<Response [200]>
met@met-All-Series:~/Documenti/GitHub/Progetto-Net2/ryu_api_solution$
```



met@met-All-Series: ~/Documenti/GitHub/Progetto-Net2/ryu_api...

```
--- 10.0.0.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 24.359/24.505/24.781/0.194 ms
mininet> link s2 s5 down
mininet> h5 ping h7
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
^C
--- 10.0.0.7 ping statistics ---
28 packets transmitted, 0 received, 100% packet loss, time 27633ms

mininet> h5 ping h7
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=1 ttl=64 time=35.3 ms
64 bytes from 10.0.0.7: icmp_seq=2 ttl=64 time=34.5 ms
64 bytes from 10.0.0.7: icmp_seq=3 ttl=64 time=34.5 ms
64 bytes from 10.0.0.7: icmp_seq=4 ttl=64 time=34.5 ms
64 bytes from 10.0.0.7: icmp_seq=5 ttl=64 time=34.4 ms
64 bytes from 10.0.0.7: icmp_seq=6 ttl=64 time=34.4 ms
64 bytes from 10.0.0.7: icmp_seq=7 ttl=64 time=34.4 ms
^C
--- 10.0.0.7 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6007ms
rtt min/avg/max/mdev = 34.405/34.583/35.339/0.310 ms
mininet>
```

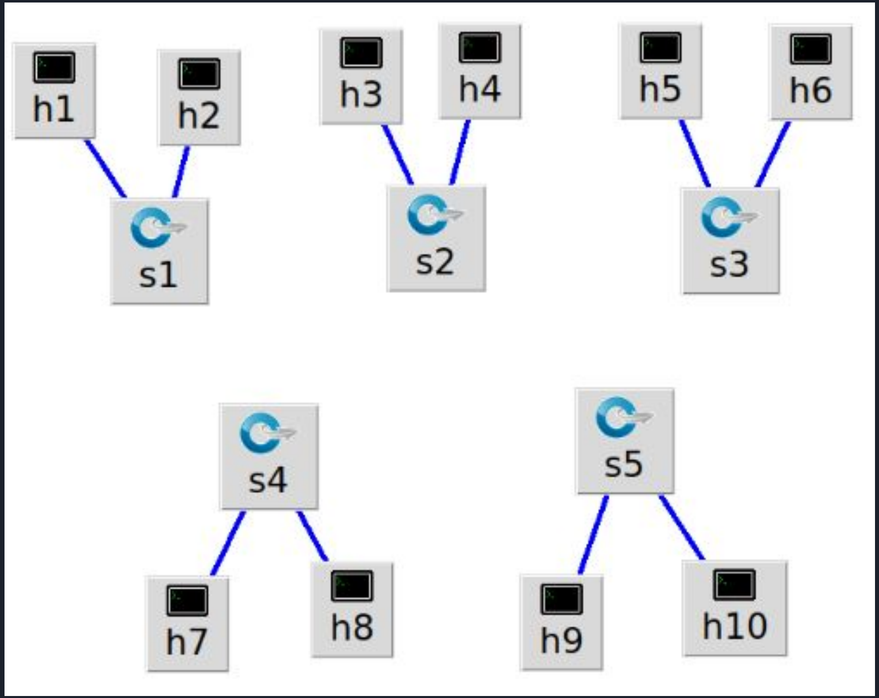
Future implementations



- ❖ Quality Of Service
- ❖ Automation

Morphing Network

Initial Network configuration



- Switches → 5
- Hosts → 10

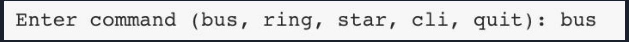
```
Host:~ $sudo python3 morph_network.py
```

```
Host:~ $ryu-manager morph_controller.py
```

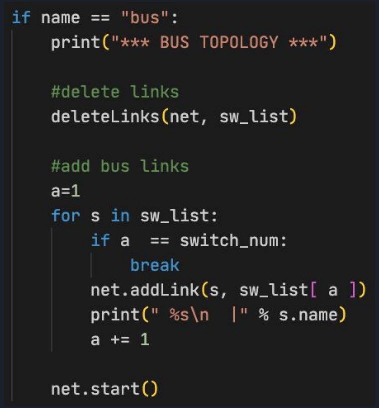
How ?

Three type of Topology

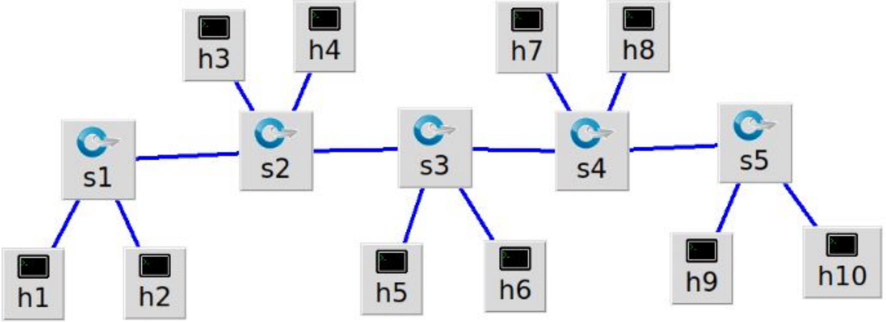
- *Bus Topology*



```
Enter command (bus, ring, star, cli, quit): bus
```

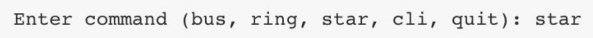


```python
if name == "bus":
    print("*** BUS TOPOLOGY ***")

    #delete links
    deleteLinks(net, sw_list)

    #add bus links
    a=1
    for s in sw_list:
        if a  == switch_num:
            break
        net.addLink(s, sw_list[ a ])
        print(" %s\n  |" % s.name)
        a += 1

    net.start()
```

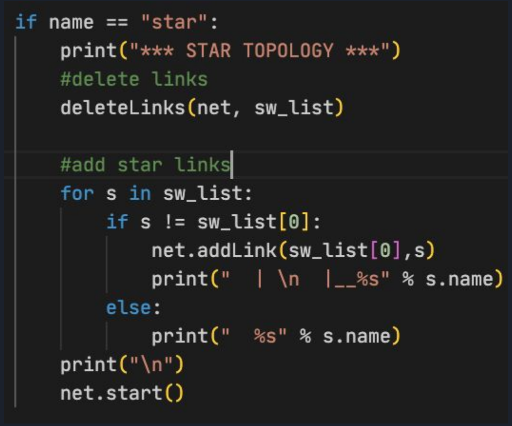


Three type of Topology

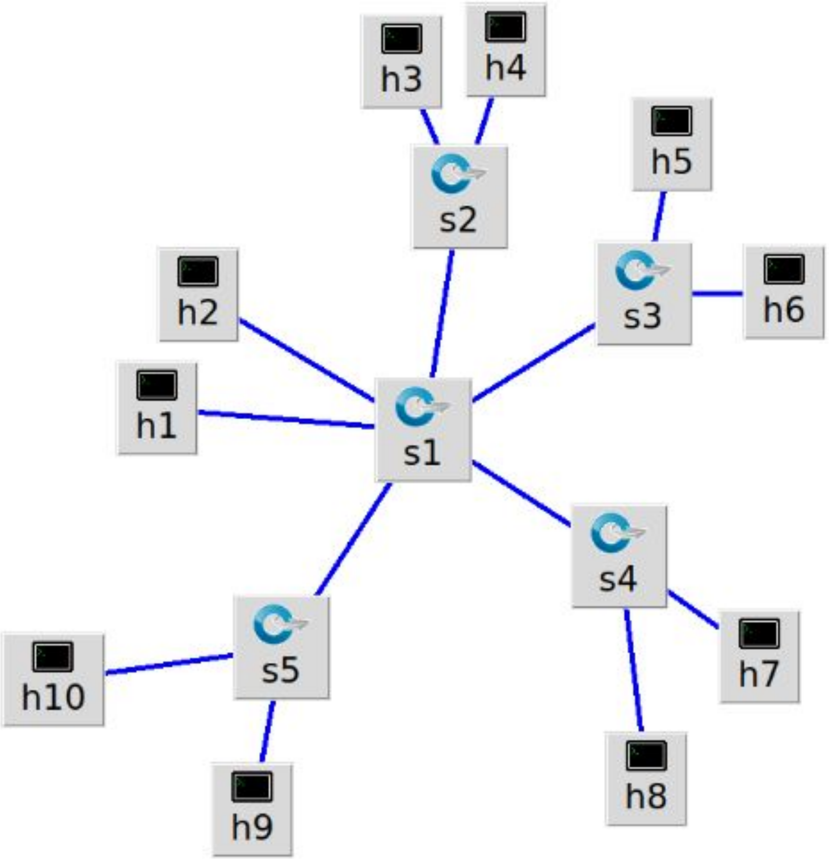
- *Star Topology*



```
Enter command (bus, ring, star, cli, quit): star
```

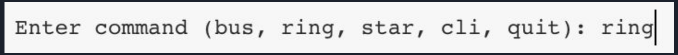


```
if name == "star":
    print("*** STAR TOPOLOGY ***")
    #delete links
    deleteLinks(net, sw_list)

    #add star links
    for s in sw_list:
        if s != sw_list[0]:
            net.addLink(sw_list[0],s)
            print("  | \n  |__%s" % s.name)
        else:
            print("  %s" % s.name)
    print("\n")
    net.start()
```

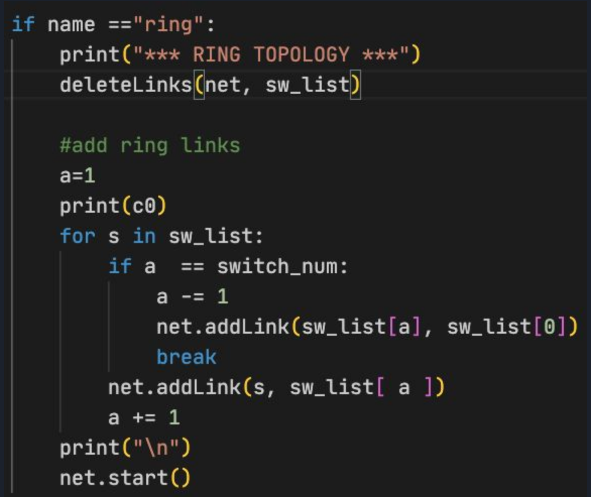


Three type of Topology

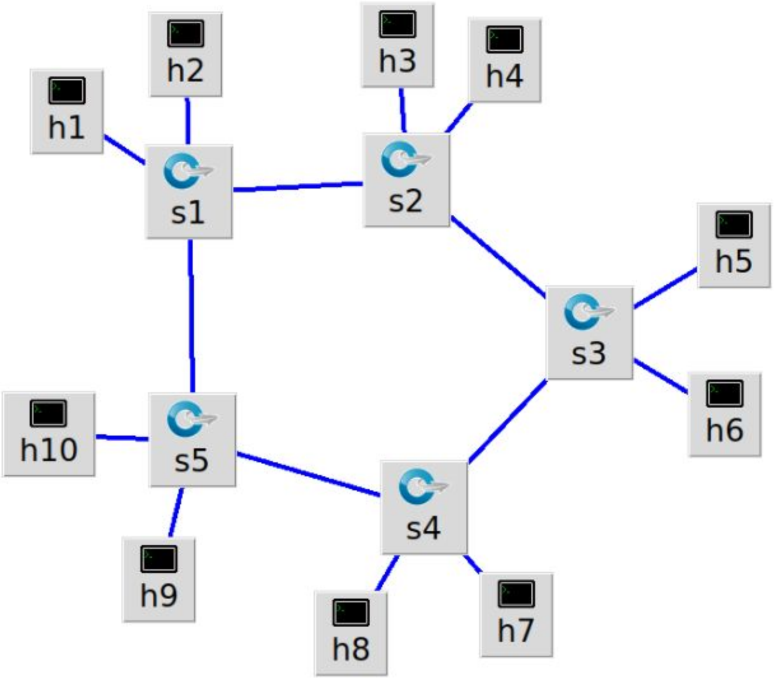
- *Ring Topology*



```
Enter command (bus, ring, star, cli, quit): ring
```



```python
if name =="ring":
    print("*** RING TOPOLOGY ***")
    deleteLinks(net, sw_list)

    #add ring links
    a=1
    print(c0)
    for s in sw_list:
        if a  == switch_num:
            a -= 1
            net.addLink(sw_list[a], sw_list[0])
            break
        net.addLink(s, sw_list[ a ])
        a += 1
    print("\n")
    net.start()
```

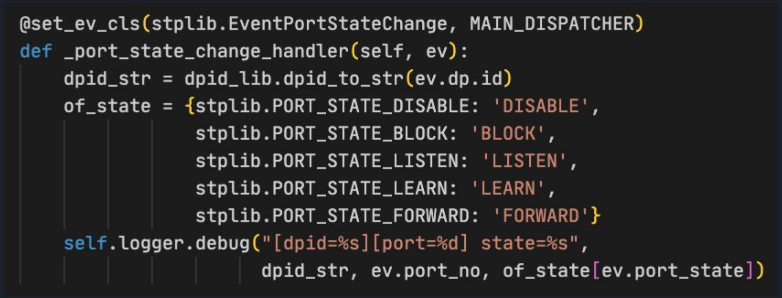


Problems and Solutions

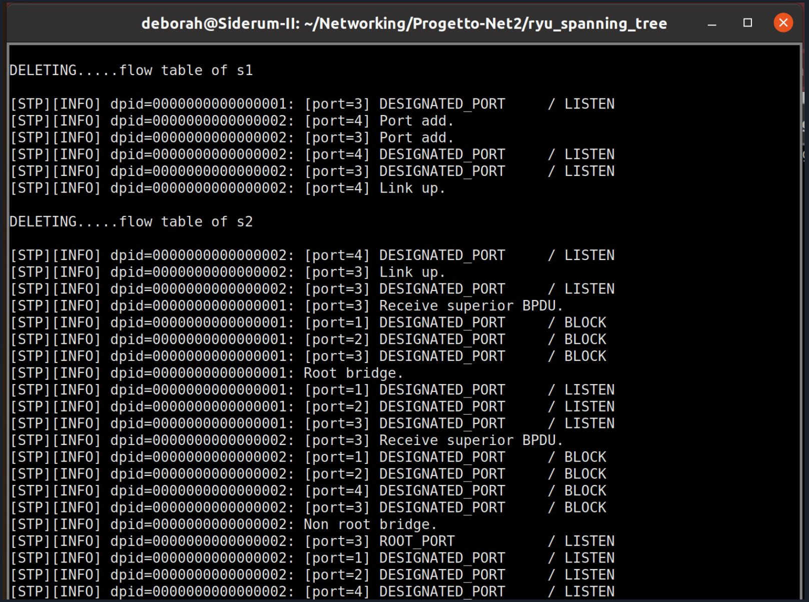
1 In the Ring Topology is present the problem of a loop in the Network

↓

The solution is that of using the Spanning Tree Protocol where all switch ports are categorized



```
@set_ev_cls(stplib.EventPortStateChange, MAIN_DISPATCHER)
def _port_state_change_handler(self, ev):
    dpid_str = dpid_lib.dpid_to_str(ev.dp.id)
    of_state = {stplib.PORT_STATE_DISABLE: 'DISABLE',
                stplib.PORT_STATE_BLOCK: 'BLOCK',
                stplib.PORT_STATE_LISTEN: 'LISTEN',
                stplib.PORT_STATE_LEARN: 'LEARN',
                stplib.PORT_STATE_FORWARD: 'FORWARD'}
    self.logger.debug("[dpid=%s][port=%d] state=%s",
                      dpid_str, ev.port_no, of_state[ev.port_state])
```



deborah@Siderum-II: ~/Networking/Progetto-Net2/ryu_spanning_tree

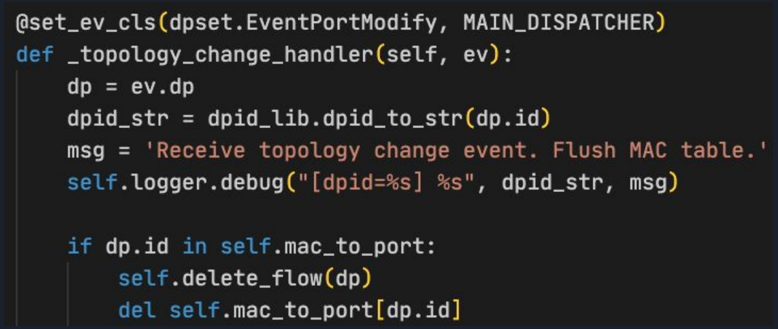
```
DELETING.....flow table of s1

[STP][INFO] dpid=0000000000000001: [port=3] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000002: [port=4] Port add.
[STP][INFO] dpid=0000000000000002: [port=3] Port add.
[STP][INFO] dpid=0000000000000002: [port=4] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000002: [port=3] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000002: [port=4] Link up.

DELETING.....flow table of s2

[STP][INFO] dpid=0000000000000002: [port=4] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000002: [port=3] Link up.
[STP][INFO] dpid=0000000000000002: [port=3] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000001: [port=3] Receive superior BPDU.
[STP][INFO] dpid=0000000000000001: [port=1] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=0000000000000001: [port=2] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=0000000000000001: [port=3] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=0000000000000001: Root bridge.
[STP][INFO] dpid=0000000000000001: [port=1] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000001: [port=2] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000001: [port=3] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000002: [port=3] Receive superior BPDU.
[STP][INFO] dpid=0000000000000002: [port=1] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=0000000000000002: [port=2] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=0000000000000002: [port=4] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=0000000000000002: [port=3] DESIGNATED_PORT     / BLOCK
[STP][INFO] dpid=0000000000000002: Non root bridge.
[STP][INFO] dpid=0000000000000002: [port=3] ROOT_PORT           / LISTEN
[STP][INFO] dpid=0000000000000002: [port=1] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000002: [port=2] DESIGNATED_PORT     / LISTEN
[STP][INFO] dpid=0000000000000002: [port=4] DESIGNATED_PORT     / LISTEN
```

Problems

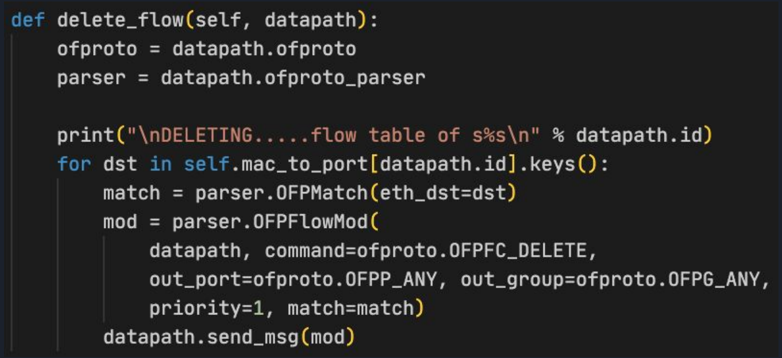
2 When there is a change in the Links of the Network the Switches doesn’t automatically delete the their Mac table

↓

The solution is that of adding a new event handler to the Ryu controller so that it deletes them when detect a topology change



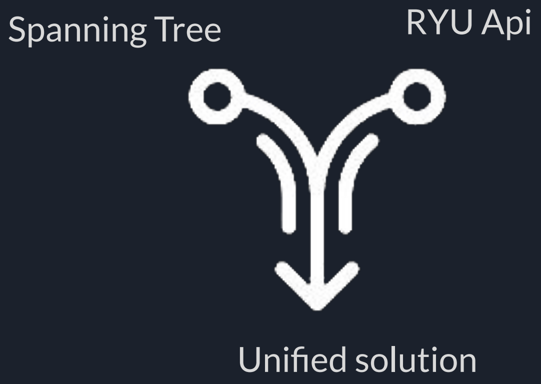
```
@set_ev_cls(dpset.EventPortModify, MAIN_DISPATCHER)
def _topology_change_handler(self, ev):
    dp = ev.dp
    dpid_str = dpid_lib.dpid_to_str(dp.id)
    msg = 'Receive topology change event. Flush MAC table.'
    self.logger.debug("[dpid=%s] %s", dpid_str, msg)

    if dp.id in self.mac_to_port:
        self.delete_flow(dp)
        del self.mac_to_port[dp.id]
```



```
def delete_flow(self, datapath):
    ofproto = datapath.ofproto
    parser = datapath.ofproto_parser

    print("\nDELETING.....flow table of s%s\n" % datapath.id)
    for dst in self.mac_to_port[datapath.id].keys():
        match = parser.OFPMatch(eth_dst=dst)
        mod = parser.OFPFlowMod(
            datapath, command=ofproto.OFPFC_DELETE,
            out_port=ofproto.OFPP_ANY, out_group=ofproto.OFPG_ANY,
            priority=1, match=match)
        datapath.send_msg(mod)
```

Future Implementations

Join the two solution to a single one



Spanning Tree

RYU Api

Unified solution

Thank you for your attention.