SDN Project 1

Comunicações Móveis

universidade
de aveiro

Prof. Daniel Corujo and Prof. Francisco Fontes
André Clérigo 98485
João Amaral 98373
Pedro Rocha 98256
Group 9

Tutorials

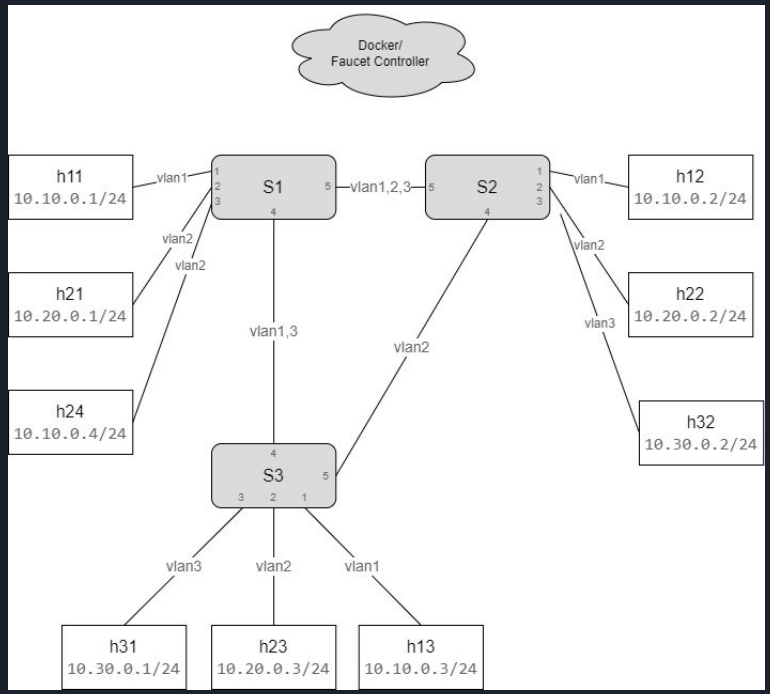
- OVS Faucet Tutorial:
  - Switching: OpenFlow tables handle packets differently
  - Routing: Packets addressed to network
  - ACLs: Blocking IPv4 TCP packets to port 8080
- IPSec Tutorial:
  - Encrypted tunnel with authentication
- Conntrack:
  - Problems setting up
- Advanced Features:
  - VLAN input and output processing, learning, look ups

Faucet - VLANs

faucet.yaml

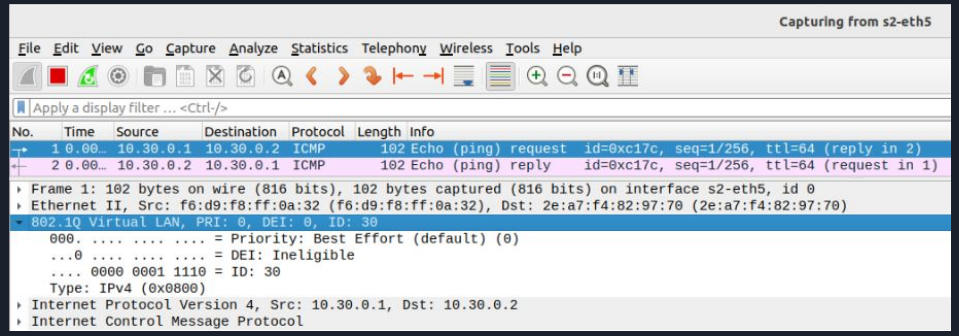
```yaml
dps:
  s1:
   description: s1
   dp_id: 0x0000000000000001
   interfaces:
      1:
          description: Connection s1 to h11
          native_vlan: 10
      2:
          description: Connection s1 to h21
          native_vlan: 20
      3:
          description: Connection s1 to h24
          native_vlan: 20
      4:
          description: Connection s1 to s3
          tagged_vlans: [10, 30]
      5:
          description: Connection s1 to s2
          tagged_vlans: [10, 20, 30]
  s2:
    description: s2
    dp_id: 0x0000000000000002
    interfaces:
      1:
        description: Connection s2 to h12
        native_vlan: 10
      2:
        description: Connection s2 to h22
        native_vlan: 20
      3:
        description: Connection s2 to h32
        native_vlan: 30
      4:
        description: Connection s2 to s3
        native_vlan: 20
      5:
        description: Connection s2 to s1
        tagged_vlans: [10, 20, 30]
  s3:
    description: s3
    dp_id: 0x0000000000000003
    interfaces:
      1:
        description: Connection s3 to h13
        native_vlan: 10
      2:
        description: Connection s3 to h23
        native_vlan: 20
      3:
        description: Connection s3 to h31
        native_vlan: 30
      4:
        description: Connection s3 to s1
        tagged_vlans: [10, 30]
      5:
        description: Connection s3 to s2
        native_vlan: 20

vlans:
  10:
    name: VLAN10
    faucet_vips: ["10.10.0.254/24"]
  20:
    name: VLAN20
    faucet_vips: ["10.20.0.254/24"]
  30:
    name: VLAN30
    faucet_vips: ["10.30.0.254/24"]

routers:
  router-1:
    vlans: [10, 20, 30]
```

Network topology

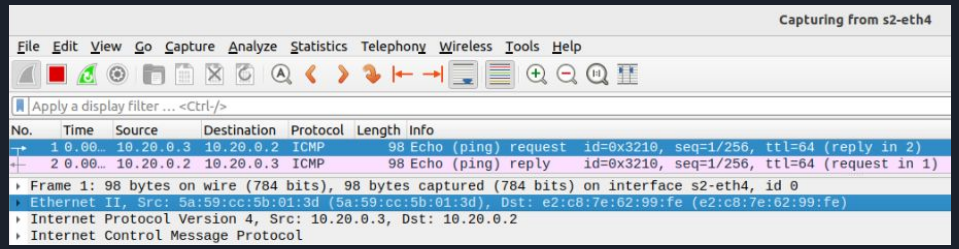


Faucet - VLANs

h31 ping h32



h23 ping h22




```
andre@cm-tutorial:~/examples$ sudo python3 mn-vlans.py
*** Creating network
*** Adding controller
*** Adding hosts:
h11 h12 h13 h21 h22 h23 h24 h31 h32
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h11) (s1, h21) (s1, h24) (s1, s2) (s1, s3) (s2, h12)
*** Configuring hosts
h11 h12 h13 h21 h22 h23 h24 h31 h32
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h11 -> h12 h13 X X X X X
h12 -> h11 h13 X X X X X
h13 -> h11 h12 X X X X X
h21 -> X X X h22 h23 X X
h22 -> X X X h21 h23 X X
h23 -> X X X h21 h22 X X
h24 -> X X X X X X X
h31 -> X X X X X X X h32
h32 -> X X X X X X X h31
*** Results: 80% dropped (14/72 received)
mininet>
```

Faucet - ACLs

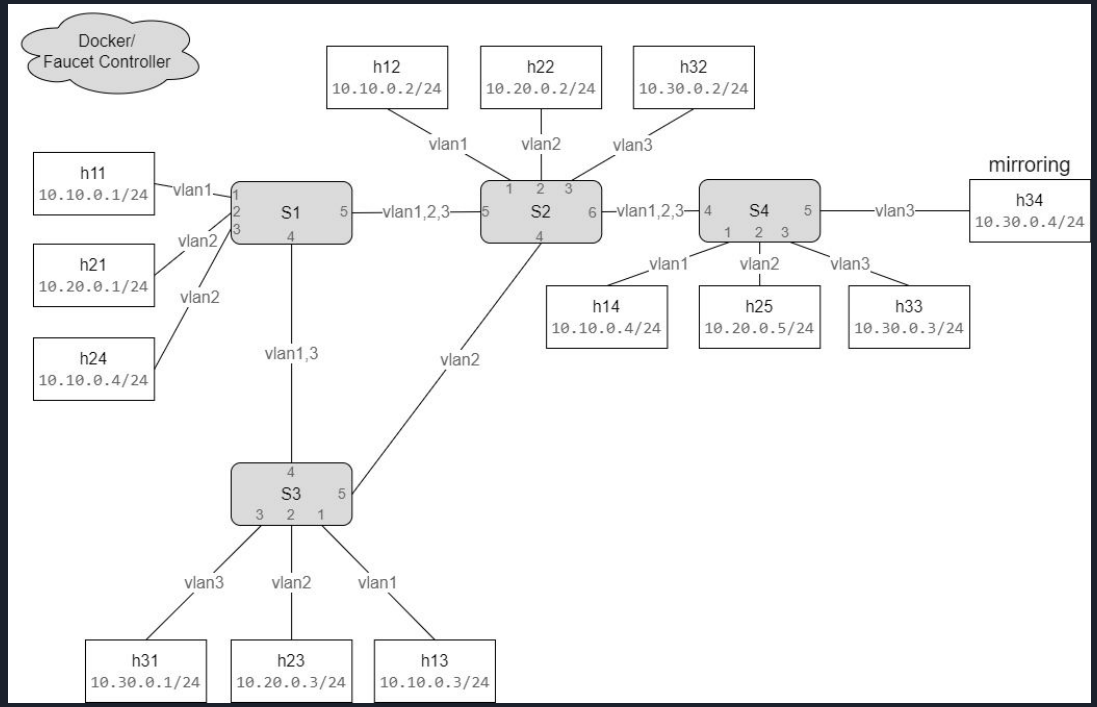
acls.yaml

```
acls:
  block-ping:
    - rule:
      dl_type: 0x800      # IPv4
      ip_proto: 1         # ICMP
      actions:
        allow: False
        mirror: 5
    - rule:
      actions:
        allow: True

# Blocking ip from malicious IP
  block-ip:
    - rule:
      dl_type: 0x800
      ipv4_src: 10.20.0.0/24
      actions:
        allow: False
        mirror: 5
    - rule:
      actions:
        allow: True

# Prioritize traffic of an IP (QoS)
# priority-traffic:
#   - rule:
#     nw_src: 1.2.3.4
#     actions:
#       priority: 100
# block-p2p:
#   - rule:
#     dl_type: 0x800
#     ip_proto: 6
#     tp_dst: [6881, 6889]
#     actions:
#       allow: False
#   - rule:
#     dl_type: 0x800
#     ip_proto: 17
#     tp_dst: [4662]
#     actions:
#       allow: False
#     - rule:
#       actions:
#         allow: True
```

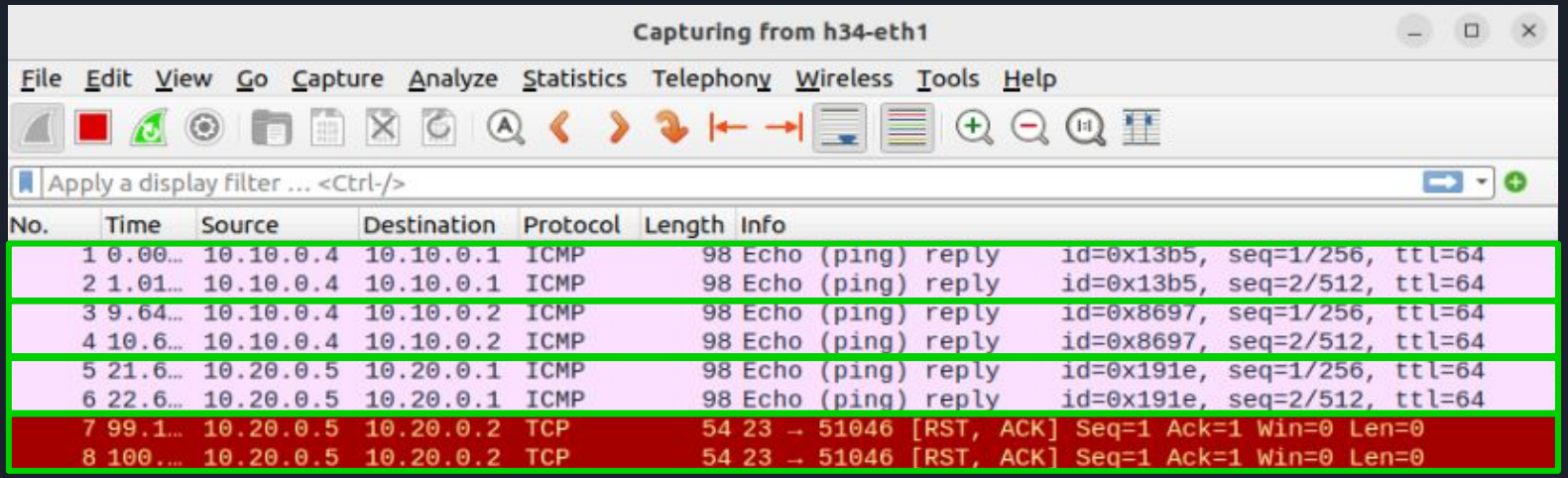
faucet.yaml

```
s4:
  description: s4
  dp_id: 0x0000000000000004
  interfaces:
    1:
      description: Connection s4 to h14
      native_vlan: 10
      acls_in: [block-ping]
    2:
      description: Connection s4 to h25
      native_vlan: 20
      acls_in: [block-ip]
```

Network topology



Faucet - ACLs



1. h11 ping h14
2. h12 ping h14
3. h21 ping h25
4. On h22 telnet 10.20.0.5

Faucet - ACLs & Monitorization

We designed a packet logger based on the ACLs introduced before:

```
(...)
[2023-01-04 18:17:20.507] - [TCP | src: 10.20.0.5      | dst: 10.20.0.2      | length: 54    | protocol:TCP]
[2023-01-04 18:17:21.507] - [TCP | src: 10.20.0.5      | dst: 10.20.0.2      | length: 54    | protocol:TCP]
[2023-01-04 18:16:43.075] - [ICMP | src: 10.10.0.4     | dst: 10.10.0.1      | length: 98    | protocol:None]
[2023-01-04 18:16:44.100] - [ICMP | src: 10.10.0.4     | dst: 10.10.0.1      | length: 98    | protocol:None]
[2023-01-04 18:16:51.876] - [ICMP | src: 10.10.0.4     | dst: 10.10.0.2      | length: 98    | protocol:None]
[2023-01-04 18:16:59.828] - [ICMP | src: 10.20.0.5     | dst: 10.20.0.1      | length: 98    | protocol:None]
[2023-01-04 18:17:00.836] - [ICMP | src: 10.20.0.5     | dst: 10.20.0.1      | length: 98    | protocol:None]
[2023-01-04 18:17:01.860] - [ICMP | src: 10.20.0.5     | dst: 10.20.0.1      | length: 98    | protocol:None]
[2023-01-04 18:17:20.507] - [TCP | src: 10.20.0.5      | dst: 10.20.0.2      | length: 54    | protocol:TCP]
[2023-01-04 18:17:21.507] - [TCP | src: 10.20.0.5      | dst: 10.20.0.2      | length: 54    | protocol:TCP]
[2023-01-04 18:17:23.524] - [TCP | src: 10.20.0.5      | dst: 10.20.0.2      | length: 54    | protocol:TCP]
(...)
```

This logger is running on the host h34, receiving blocked ping packets and packets to the h25's network

Faucet - SDN App

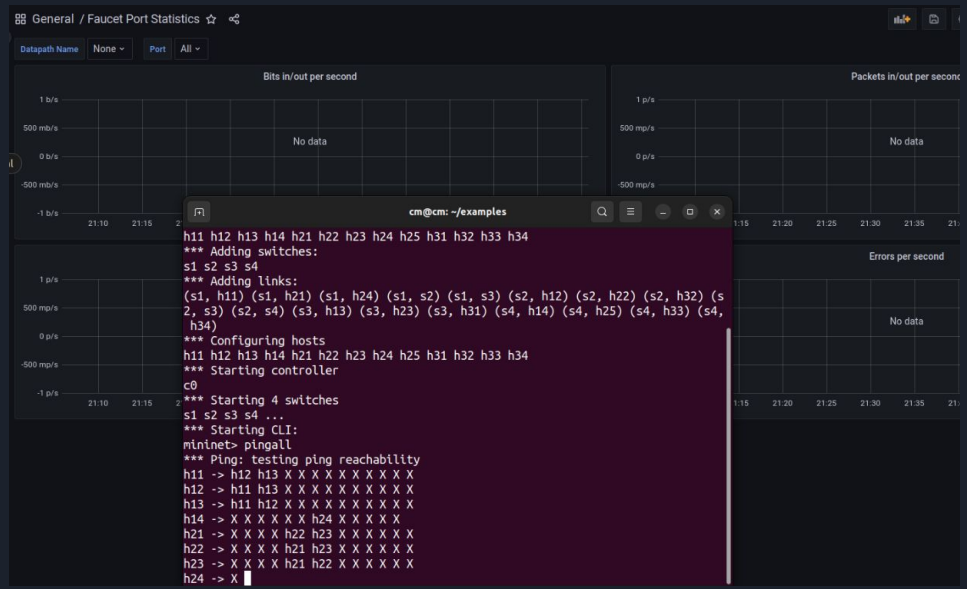
- There is no API Documentation
- Tried creating a SDN application
- Tried creating a SDN application using an “API”

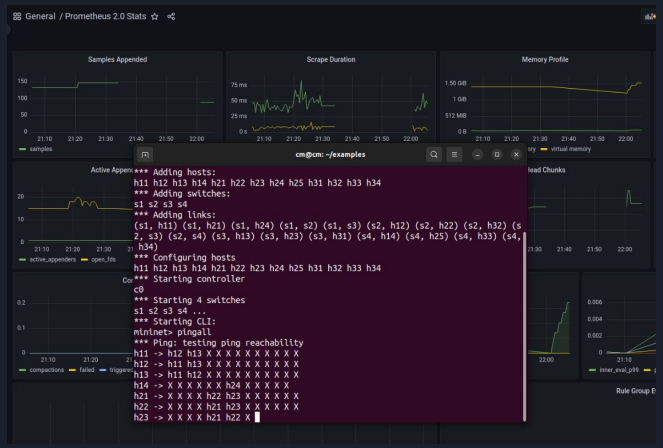
```
# Start the Mininet network
net.start()

def traffic_monitor(h2):
    while True:
        traffic = h2.cmd('ifstat -i h2-eth1 1 1')
        try:
            traffic = float(traffic.split()[-2])
        except:
            traffic = 0

        if traffic > 1:
            info(traffic)
            info('\n')
        if traffic > 50:
            info('aconteceu\n')
            s1.cmd('ovs-ofctl add-flow s1 priority=15,in_port=1,actions=drop')

# Start the traffic monitor in a separate thread
thread = threading.Thread(target=traffic_monitor, args=(h2,))
thread.start()

# Enter the Mininet CLI to interact with the network
CLI(net)
```

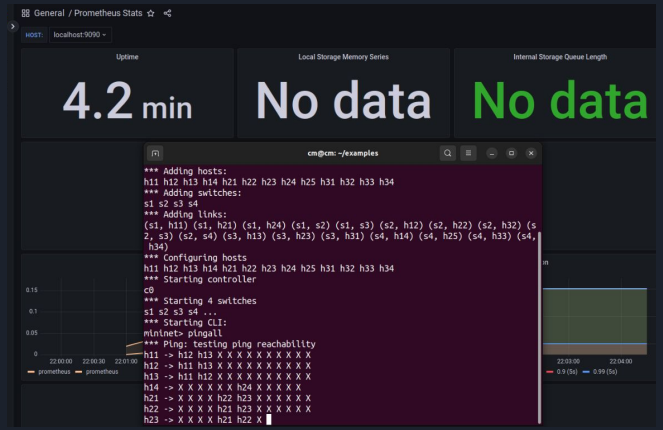
```
# Set the parameters for the traffic management configuration
data = {
    "dp_id": "0x1",
    "interfaces": [
        {
        "name": "eth1",
        "native_vlan": 1
        }
    ]
}

api_url = "http://localhost:9302/api/v1/config"
#response = requests.post(api_url, json=data)
#print(response.status_code)

api_url = "http://localhost:9302/api/v1/switches"
params = {
    "dp_id": "0x1"
}
response = requests.get(api_url, params=params)
print(response.text)
print(response.status_code)

net.start()

CLI(net)
```

Faucet - Dashboard

- Tutorial for Faucet Dashboard doesn't work
- Github Issue (https://github.com/faucetsdn/faucet/issues/4229)







GNS3 - Setup

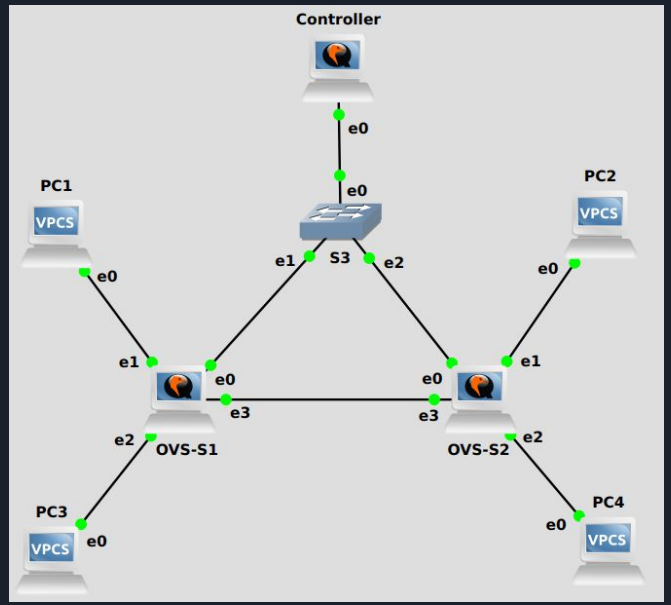
- Controller and SDN Switches use Ubuntu Server (5GB HDD, 1GB RAM)
- Controller with Faucet and ONOS

Configures done:

configure.sh
ifconfig ens [3 ... 8] up
ovs-vsctl set-controller br0 tcp:192.168.0.1:6653
ovs-vsctl add-port br0 ens [3 … 8]

openvswitch-settings.service
[Unit]
…
[Service]
Type=oneshot
ExecStart=/bin/bash /home/controller/configure.sh
[Install]
WantedBy=multi-user.target

Topology



ONOS Controller

- Basic testing of ONOS (Reactive Forwarding)
- Backup controllers

Hybrid Networks

- This is in the domain of scientific research!

In this scenario during the link discovery process, the legacy switch interrupts the propagation of LLDP packets, which is causing the controller to be unable to discover the rest of the network. There are a few potential solutions you could consider in this situation:

If the legacy switch supports LLDP packets, we could try to determine whether the switch is simply passing the packets through without interacting with them, or if there is some way that the switch can be "seen" by the controller.

We could modify the controller by modifying the existing link discovery app or developing a new app that is able to support the necessary modifications for one of the above scenarios, or you could think of a new approach to solving the problem.

Topology