SDN Practice: OpenFlow Flow Rule Management

Overview

This practice demonstrates how to set up and manage OpenFlow flow rules in a Software-Defined Network using Mininet, Open vSwitch (OVS), and the Ryu controller. The topology consists of three switches in a linear arrangement, each connected to one host.

Prerequisites

- Mininet
- Open vSwitch (OVS)
- Ryu Controller
- Python 3.x
- curl

Network Topology

The network consists of:

- 3 OpenFlow switches (s1, s2, s3) arranged linearly
- 3 hosts (h1, h2, h3), each connected to one switch
- A Ryu controller

Connection details:

- h1 ↔ s1-eth1
- h2 ↔ s2-eth1
- h3 ↔ s3-eth1
- s2-eth2 ↔ s1-eth2
- s3-eth2 ↔ s2-eth3

Setup Instructions

1. Start the Ryu controller:

```
ryu-manager ryu.app.ofctl_rest
```

2. Create the Mininet topology:

```
sudo mn --controller=remote,ip=127.0.0.1 --mac --switch=ovsk,protocols=OpenFlow13 --topo=linear,3
```

Understanding Port Mappings

You can view port mappings using:

```
sudo ovs-ofctl -O OpenFlow13 dump-ports <switch_name>
```

Important port mappings:

- Switch 1 (s1): eth1 → h1, eth2 → s2
- Switch 2 (s2): eth1 → h2, eth2 → s1, eth3 → s3
- Switch 3 (s3): eth1 → h3, eth2 → s2

Flow Rule Management

Flow Rule Structure

Basic structure of a flow rule JSON:

```
{
    "dpid": <switch_id>,
    "table_id": 0,
    "priority": 100,
    "match":{
        "eth_dst": "<MAC_address>"
    },
    "actions":[
        {
            "type":"OUTPUT",
            "port": <port_number>
        }
    ]
}
```

Adding Flow Rules

Use curl to add flow rules:

```
curl -X POST http://localhost:8080/stats/flowentry/add -d @flow_rule.json
```

Viewing Flow Rules

Check existing flow rules:

```
curl http://localhost:8080/stats/flow/<switch_id>
```

Or using OVS directly:

```
sudo ovs-ofctl -O OpenFlow13 dump-flows <switch_name>
```

Common Issues and Solutions

1. **Missing Bidirectional Rules**: Remember to add flow rules for both directions of traffic.
    - Example: For h2 to ping h3, you need rules in:
        - s2 for traffic to h3
        - s3 for traffic to h2
2. **Port Mapping Confusion**: Use `dump-ports` to verify correct port numbers:

```
sudo ovs-ofctl -O OpenFlow13 dump-ports <switch_name>
```

3. **ARP Handling**: Include ARP flow rules to handle address resolution:

```
{
    "match":{"dl_type": 2054},
    "actions":["OUTPUT:FLOOD"]
}
```

Testing Connectivity

Test connectivity using ping:

```
mininet> h1 ping h2
mininet> h2 ping h3
```

Troubleshooting Tips

1. Verify switch connections to the controller
2. Check flow rules on all switches in the path
3. Verify port mappings using `dump-ports`
4. Ensure ARP rules are in place
5. Check both directions of traffic flow

Additional Notes

- Always use OpenFlow 1.3 (`-O OpenFlow13`) when working with OVS commands
- Remember to add flow rules for both forward and return traffic
- Use appropriate port numbers based on the topology
- Consider ARP traffic handling in your flow rules