SDN Lab: Multi-table Flow Implementation

Problem Description

Implement a software-defined network using Mininet and Ryu controller with multiple flow tables to manage network traffic policies.

Network Topology

- 3 OpenFlow switches (s1, s2, s3) connected linearly
- 4 hosts:
  - h1 (10.0.0.1) and h2 (10.0.0.2) connected to s1
  - h3 (10.0.0.3) and h4 (10.0.0.4) connected to s3

Requirements

Custom Topology

Create a custom topology python script using Mininet API with the above specifications.

Ryu Controller Application

Implement a Ryu application that:

1. Uses OpenFlow 1.3
2. Implements multiple flow tables:
   - Table 0: Source IP and packet type verification
   - Table 1: ICMP traffic management
   - Table 2: TCP traffic control (ports 80, 443)
   - Table 3: Default forwarding

Traffic Policies

- Block ICMP traffic between h1 and h4
- Allow only TCP ports 80 and 443
- ARP packets must flood

Validation Tips

1. Check flow entries in all switches:

```
sudo ovs-ofctl -O OpenFlow13 dump-flows s1
sudo ovs-ofctl -O OpenFlow13 dump-flows s2
sudo ovs-ofctl -O OpenFlow13 dump-flows s3
```

2. Test connectivity:

```
h2 ping h4  # Should work
h1 ping h4  # Should fail
h2 iperf h4 # Should work on allowed ports
```