Module Title: Software-Defined and Wireless Network Security

Assignment number: 1

Assignment Topic: SDN and simple firewall design

Submission deadline: 9th November 2025

Total marks: 100 (40+60)

Weightage: 15%

Objective:

- The first objective of the assignment is to become familiar with the reasons for selecting appropriate technology from SDN, Network slicing, and NFV.
- The second objective of the assignment is to show a controller-based network that can be designed and implemented in a Mininet.
- How a simple firewall in a Software-defined network can be implemented.

NOTE:

Complete only one of the following.

- Part 1 and Part 2 as an assignment task of 100 marks
- Or
- Challenge part only, which is also of 100 marks.

Part 1: Network Design (Topology + Controller Setup) (40 Marks)

- Use Mininet to design a custom topology (minimum 6 switches, 1 or 2 controllers, and at least 6 hosts).
- Include redundant links or multiple paths between sub-networks.
- Label hosts as belonging to different “departments” or “services” (e.g., Admin, Students, IoT).
- Connect the topology to a Ryu or POX controller.
- Configure OpenFlow version 1.3 or later.

Deliverable:

A report (PDF) discussing all the above points. Maximum 1000 - 1500 words. Include screenshots in your PDF report as well, and also provide the code file.

Marking:

- Topology creation and implementation: 10 marks.
- Multiple paths or redundant links: 10 marks
- Controller connectivity: 10 marks
- OpenFlow and testing: 10 marks
- File is mandatory.

Part 2: Implement Secure SDN Flows (60 Marks)

Implement and demonstrate any **three** of the following security features using OpenFlow rules or controller logic. Create a simple topology like in part 1 for this task.

1. Firewall Functionality:
   - Block specific traffic (e.g., deny ping or TCP from one subnet to another).
   - Permit only defined flows.
2. DDoS Detection/Mitigation:
   - Monitor packet-in rate per host; if traffic exceeds the threshold, block or limit that host.
3. Access Control Lists (ACL):
   - Only allow traffic between certain VLANs or IP ranges.
4. Dynamic Routing Policy:
   - Use the Ryu app or POX script to change routes based on link utilisation or switch status.

Deliverables:

- Python script.
- Flow table dump showing applied rules (dpctl dump-flows)
- Short explanation of logic (what is being blocked/allowed and why)
- A report discussing the design and the implementation. Use a screenshot of each step and show it's working. (max 1000-1500 words)
- Provide two separate Python code files for each of the above stages, with and without a simple firewall (For task 1).

Marking:

- Each part carries 20 marks.
- 10 marks for the code correctness and working.
- 5 marks for the logical explanation.
- 5 marks for testing.
- File is mandatory.

** If you fail to do that, it is still possible to get at least some marks. If a proper reason is given as to why the selected approach didn’t work, and if your approach is correct.

** Refer to course labs for controller-based topology creation, flows and firewall.

**** This is the challenge part.**

Challenge part (100 marks)

Demonstrate the use of FlowVisor to create multiple virtual network slices on a shared physical topology. Each slice should be managed by a separate controller (e.g., Ryu and POX) and isolated in terms of traffic control.

Tasks

1. **Install and Configure FlowVisor**
   - Install FlowVisor in your Mininet environment.
   - Set up FlowVisor to act as a proxy between the OpenFlow switches and multiple controllers.
2. **Define Slices**
   - Create at least two slices:
     1. Slice 1 (e.g., Admin/Control) – controlled by Controller A
     2. Slice 2 (e.g., Student/Data) – controlled by Controller B
   - Assign different switches or flowspaces to each slice using FlowVisor's fvctl commands.
3. **Implement Basic Policies**
   - Each controller should implement simple, distinct logic (e.g., Slice 1 = allow ICMP only, Slice 2 = TCP only).
   - Verify that one controller cannot affect the other slice.
4. **Measure Isolation and Performance**
   - Use ping and iperf tests to compare:
     1. Latency and throughput within each slice
     2. Traffic isolation (no cross-slice interference)
   - Capture results and briefly discuss performance and slice isolation.

Deliverables:

A report containing the explanation and description of the steps for completing the above tasks. The word count is 1500 to 200 words. Include screenshots in your report with an explanation. Provide the code file as well.

Marks

- FlowVisor configuration: 30 marks (correct slice setup and controller working)
- Isolation verification: 30 marks (cross slice isolation clearly shown)
- Measurements and testing: 20 marks (collected and analysed correctly)
- Documentation: 20 marks (clear explanation and screenshots)
- File is mandatory.

LINKS:

https://github.com/onstutorial/onstutorial/wiki/Flowvisor-Exercise

https://github.com/opennetworkinglab/flowvisor

https://github.com/syaifulahdan/mininet/blob/master/mininet/flowvisor-instalation.md

https://orbit-lab.org/wiki/Internal/OpenFlow/Controllers/FlowVisor