Name Bin Mohsin, Taaha and He, Haiyi Date: 05/16/2025

(last name, first name)

NYU ID: N15200048 and <insert Haiyi’s N number>

Course Section: 001

**Assignment 8: Final Project**

\_\_\_\_\_ Part 3 functionality [Max 70 points]

\_\_\_\_\_ Part 4 functionality [Max 25 points]

\_\_\_\_\_ Style [Max 5 points]

\_\_\_\_\_ Part 3 extra credit questions [Max 20 points (10 each)]

**\_\_\_\_\_ Total [Max 100 points + Extra Credit points]**

**Professor’s Comments:**

**Affirmation of my Independent Effort:** A signature on a white background

AI-generated content may be incorrect. <insert Haiyi’s signature>

**(Sign here)**

**Title: SDN-Based Shortest Path Switching and Distributed Load Balancer using Floodlight**

**Abstract**

This report documents the design and implementation of a Software-Defined Networking (SDN) project developed using the Floodlight controller. The project consists of two major components: (1) a Shortest Path Switching (SPS) module that ensures efficient unicast forwarding across a dynamic network topology, and (2) a Load Balancer (LB) module that intercepts TCP connections to a virtual IP and redistributes them across a pool of backend servers. The work aims to demonstrate the flexibility and power of SDN for both control and application-layer functionalities.

**1. Introduction**

Traditional networking often suffers from rigid control planes and decentralized routing logic. SDN addresses this by centralizing network control logic, allowing smarter, adaptive forwarding policies. In this project, we leveraged the Floodlight controller to build a modular SDN application consisting of:

* A shortest-path unicast switching module using controller-computed paths
* A TCP-aware load balancer that rewrites and forwards traffic to backend hosts via VIPs

The system is designed to react dynamically to host mobility, link failures, and topology changes while maintaining correctness and performance.

**2. Related Work**

Shortest-path forwarding has long been a core problem in networking, traditionally handled by distributed protocols like OSPF. In SDN, the controller's global view simplifies this computation. Similarly, load balancing at Layer 4 is typically performed by appliances or proxies; SDN allows for fine-grained rule management without requiring inline middleboxes. Prior SDN-based systems like OpenDaylight and ONOS have demonstrated similar concepts; however, our implementation is purpose-built for educational use atop Floodlight.

**3. System Architecture and Algorithms**

**3.1 Shortest Path Switching (SPS)**

The SPS module maintains a network graph using link discovery events and computes shortest paths using a BFS variant of Dijkstra's algorithm. When a host joins or moves, rules are installed:

* In the switch to which the host is directly connected (IP- and MAC-based)
* On all other switches along the shortest path to that host

Rules include:

* ARP packets sent to the controller
* IPv4/MAC destination matches directing traffic to the next hop

**3.2 Load Balancer**

The LB module is placed in Table 0 and intercepts:

* All ARP packets destined for VIPs
* TCP SYN packets targeting VIPs

It selects a backend via round-robin and installs two rewrite rules:

* From client to backend (matching on VIP, rewriting IP/MAC)
* From backend to client (matching on backend IP, rewriting back to VIP)

Packets are then forwarded to Table 1 (SPS) for final delivery.

**4. Implementation**

The application was written in Java atop the Floodlight v1.2 controller with support for OpenFlow 1.3. Key implementation decisions:

* The SPS module relies on IDevice and ILinkDiscovery services to maintain topology
* The LB module handles ARP replies and packet rewrites directly via packet-out messages
* A default rule passes all unmatched packets from Table 0 to Table 1
* Rules are installed with appropriate priority and timeouts to ensure correctness and scalability

A detailed modular breakdown is provided in the submitted source code and readme.txt.

**5. Results and Evaluation**

The project was validated using Mininet topologies with 3 to 6 hosts and various link permutations. The following behaviors were confirmed:

* Bidirectional communication between all hosts under dynamic topology changes
* Correct round-robin distribution of TCP sessions to backend servers
* Proper ARP resolution for VIPs
* Fast recovery from host mobility and link state updates

Packet captures and flow table dumps confirmed that:

* SPS rules match on either MAC or IP with correct out ports
* LB rewrite rules are installed only on client switches
* No unintended flooding occurred

**6. Conclusion**

This project demonstrates the expressive power of SDN for implementing both Layer 2/3 forwarding and Layer 4 load balancing. The modularity of the Floodlight controller enabled clean separation between the SPS and LB logic. Future work could include support for Equal Cost Multi-Path (ECMP), reactive loop-free multicast (spanning trees), and TCP state monitoring for better backend utilization metrics.

**References**

1. Floodlight Open SDN Controller: http://www.projectfloodlight.org/
2. Mininet Network Emulator: http://mininet.org/
3. "OpenFlow Switch Specification v1.3.0", ONF
4. Computer Networking: A Top-Down Approach (8th Edition)  
   James F. Kurose, Keith W. Ross