**CSC/ECE 573 (001) – Internet Protocols**

**Dynamic** **Failure Recovery in OpenFlow based Software Defined Networks**

Suraj Kumar K P Srivatsan Narasimhan Adharsh Srivats R (200270123) (200260635) (200254962)

Neha Manivannan Maitraye Murali

(200268645) (200254483)

# **1. INTRODUCTION**

Software Defined Networking (SDN) is a network architecture that decouples the control and data planes. It allows operators to manage the entire network consistently and holistically, regardless of the underlying network technology. It also facilitates users to develop network-aware applications, intelligently monitor network conditions, and automatically adapt the network configuration as needed. We use OpenFlow Architecture to control our SDN environment.

# **2. PROBLEM STATEMENT**

Fast resiliency is the network’s ability to adapt to failures and resume to normalcy. Networks have to guarantee fast resiliency upon network link failure and since OpenFlow doesn’t guarantee it, we aim to enhance the OpenFlow architecture by adding fast recovery mechanisms in the switch and the controller.

**3. OBJECTIVES**

* To enable fast recovery in OpenFlow networks.
* To use link protection scheme to compute the backup path and redirect the traffic.
* To manipulate flow tables based on the status of the link and make appropriate decisions to handle data packets.

**4. IMPLEMENTATION**

The system architecture is represented in Fig 1. The system consists of four major modules: recovery module, link failure detection and notification, renewal packet generation and working path restoration.

* The controller calculates the backup path proactively and installs it in the switch.
* The switch generates renewal packets to keep the existing backup paths alive irrespective of the idle time of flow entries
* When a link fails, the switch identifies the failed link and deletes its flow entries which follow the failed link
* Once the failed link is up, switch notifies the controller to recalculate the best path by sending recovery packets.

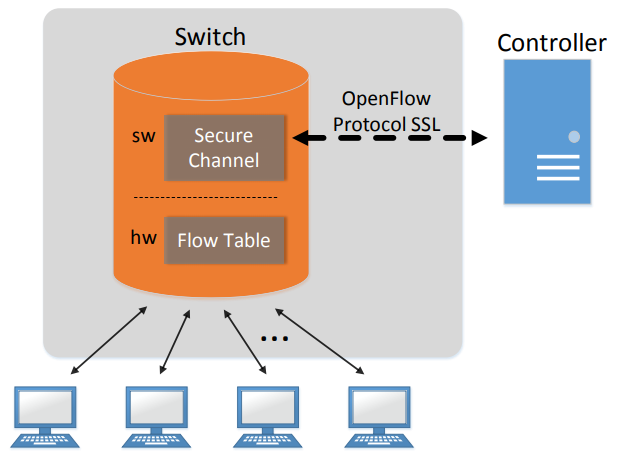


Fig 1 OpenFlow Architecture

**5. EVALUATION**

Two Linux systems will be used, one to install OpenFlow controller and another for installing Mininet and OpenvSwitch. The OpenFlow controller, switch and protocol are extended to incorporate the required changes.

The network is designed in such a way that every switch in the working path has a backup path to the destination.The performance of the system is evaluated using the number of packets lost during a continuous flow. To find this value, continuous flow of ping requests are given between the two hosts of the emulated network. Link failure is then simulated and the number of packets lost during the flow is monitored by explicitly terminating the ping request.

**6. DEMO SET-UP**

* To set up the basic OpenFlow topology. (weightage 30)
* To analyze the data path when the link is down. (weightage 40)
* To analyze the data path when the link is up.(weightage 30)