Project Title: Implementation of Secure Multipath Resilience in HIPLS using Mininet and SDN Integration

Introduction

The demand for safe and efficient communication across networks is becoming crucial in today's digital era. Traditional Internet Protocol (IP) addresses serve two primary functions: identifying devices on a network and determining their location. However, this dual role can create security challenges and limit network management flexibility. The Host Identity Protocol (HIP) is utilized in this context. HIP separates these two roles by introducing a new identification layer, allowing devices to be recognized by cryptographic identities, making it harder for unauthorized users to gain access.

Consider a large organization that maintains office locations in multiple cities. To ensure all their offices can share information as if they were on the same local network, they can use Virtual Private LAN Service (VPLS) technology. VPLS creates a virtual network that connects different locations seamlessly over the Internet, as if cables physically connect them. The combination of HIP and VPLS, known as Host Identity Protocol-based Virtual Private LAN Service (HIPLS), comes into play to make the VPLS network secure. HIPLS ensures that only authorized devices can join the network, and it allows these devices to communicate securely across different locations[1].

Additionally, managing networks effectively is crucial in a world that is becoming more complex. Software-defined networking (SDN) is an innovative approach that separates the control of the network from the physical hardware, making it easier to manage and adapt the network through software. This flexibility is critical to creating networks that can quickly respond to new challenges, such as security threats or changes in traffic patterns.

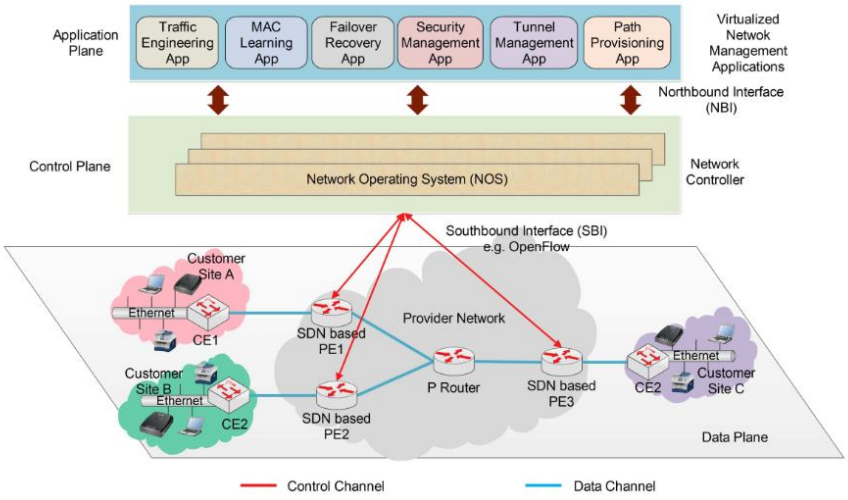
This project will explore how combining HIP, VPLS, and SDN can create a highly secure and adaptable network environment. This has real-world implications for safeguarding sensitive business information and ensuring reliable communication in critical infrastructures such as healthcare and financial institutions.

[1] For real-world scenarios, visit https://www.tempered.io/

Project Description:

This project aims to implement and evaluate a multipath resilience solution in a HIPLS environment, integrated with SDN using Mininet as the network emulator and Ryu as the SDN controller. The primary focus will be developing a robust network system capable of dynamic path failover and secure multi-party key exchange, ensuring continuous and secure communication even in network failures or possible attacks.

Figure 1: SDN-VPLS



Project Objectives:

1. **Secure Multi-party Key Exchange:**
    - Implement a multi-party key exchange protocol (e.g., Burmester-Desmedt key agreement protocols) within the HIPLS to ensure secure communication between hosts in the network.
    - Integrate the key exchange protocol with the SDN controller to dynamically manage and distribute cryptographic keys.
2. **Dynamic Path Failover:**
    - Develop and implement a path failover mechanism that detects real-time network failures and attacks and dynamically switches traffic to alternative routing paths.
    - This new feature will be added in the *Application Plane,* as depicted in Figure 1.
3. **Simulation Environment:**
    - Basic Python implementation of HIPLS can be found at: https://github.com/strangebit-io/hip-vpls
    - Use Mininet and Ryu to simulate various network scenarios, including link failures.

- Evaluate the performance of the multipath resilience mechanism in terms of failover time and throughput.
- Analyze the effectiveness of the implemented multi-party key exchange in maintaining secure communication.

Project Deliverables:

1. **Source Code:**
   - Complete basic source code for the multipath routing, dynamic path failover mechanism, and multi-party key exchange protocol.
   - Scripts and configuration files required to run the simulations in Mininet and Ryu.
2. **Technical Documentation:**
   - User manual for deploying and testing the system in different network scenarios.
   - Document the integration process between HIPLS and Ryu SDN controller.
3. **Simulation Results and Analysis:**
   - Detailed report on the performance evaluation of the implemented system.
   - Discussion of the results, highlighting the strengths and limitations of the implemented solution.

Requirements:

- Completing the Computer Networks and Distributed Systems course (TDTS04-TDTS06-TDTS11 or any related course) is essential.
- Advanced Python programming skills (SDN controller and mininet were written purely in Python).
- A basic understanding of modern networking concepts (e.g., SDN) is beneficial.