Using multiple wireless access point

1nd Le Gia Hoang Thien
22521387 - Computer Networks and Communications
Thu Duc District, HCMC

2th Le Thi Thuy Trang
22521511 - Computer Networks and Communications
Thu Duc District, HCMC

3st Le Hoang Vu
22521691 - Computer Networks and Communications
Thu Duc District, HCMC

4rd Pham Thi Thanh Vinh
22521680 - Computer Networks and Communications
Thu Duc District, HCMC

***Abstract*—In recent years, Software-Defined Networking (SDN) has emerged as a revolutionary network architecture that offers centralized control over network management. With the increasing demand for high-performance and seamless wireless services in environments such as offices, universities, and resorts, managing multiple Access Points (APs) has become a significant challenge. This paper proposes an SDN-based solution to optimize Wi-Fi networks with multiple APs, focusing on client load balancing, seamless roaming, and fault detection. The system utilizes an SDN controller to monitor and manage APs, dynamically redistribute clients, and ensure optimal performance. We demonstrate the proposed solution through simulations using Mininet-WiFi and the Ryu controller, with results showing significant improvements in network stability, performance, and maintainability.**

I. Introduction

A. Project Overview

The increasing reliance on wireless connectivity has created a demand for efficient, scalable, and manageable Wi-Fi systems, especially in large-scale environments like office buildings, campuses, resorts, and public spaces. Traditional methods of managing Wi-Fi networks often fall short in handling multiple Access Points (APs) concerning client load balancing, seamless roaming, and fault management. These problems arise due to the distributed nature of APs, where each AP operates independently without a centralized controller.

The objective of this project is to design and implement an SDN-based system that optimizes the operation of multiple APs within a network. By leveraging SDN, we can achieve centralized management of APs, enabling flexible client distribution control, real-time network status monitoring, and automated fault detection. Specifically, the system will:

- **Load Balance:**Evenly distribute clients among APs based on real-time indicators such as traffic and signal quality.
- **Enable Seamless Roaming:** Ensure that clients can move between APs without service interruption.
- **Detect and Recover from Faults:** Identify AP failures and automatically migrate clients to healthy APs.

To validate the system's effectiveness, the project employs Mininet-WiFi, an extension of Mininet for simulating wireless networks, and the Ryu SDN controller for managing data flows and client distribution. Simulation results will illustrate how SDN enhances the performance and scalability of multi-AP Wi-Fi networks [1].

B. Research Problem

Managing a large number of Access Points (APs) in a wireless network introduces several challenges:

- **Client Load Imbalance:** Some APs may become overloaded while others remain underutilized, leading to suboptimal network performance.
- **Roaming Issues:** Clients may experience service interruptions or degraded performance when transitioning between APs, especially in large environments.
- **Fault Detection:** Detecting AP failures and rerouting traffic to healthy APs without manual intervention is complex.

These issues can be addressed through SDN, where the control plane is separated from the data plane, enabling centralized network management. Using SDN to manage multiple APs allows for automated load balancing, improved roaming performance, and simplified fault detection and recovery.

C. Project Objectives

The main objectives of this project are:

1) **Design an SDN-based architecture** to optimize multi-AP Wifi networks.
2) **Implement a dynamic load balancing algorithm** to distribute clients based on real-time metrics such as client count and signal strength.
3) **Develop a fault detection system** to identify AP failures and automatically switch clients to healthy APs.
4) **Simulate the proposed system** using Mininet-WiFi and the Ryu controller to evaluate performance.
5) **Assess system performance** in terms of load balancing efficiency, seamless roaming, and fault recovery capability.

D. Scope and Limitations

This project focuses on:

- Designing and simulating a multi-AP network using SDN.
- Implementing and testing client load balancing, seamless roaming, and fault detection.

- Utilizing Mininet-WiFi and Ryu as the simulation tools.

However, it is subject to the following limitations:

- The simulation is limited to a small-scale network with a finite number of APs and clients.
- The fault detection mechanism is simplified and may not cover all real-world scenarios.
- The system is not integrated with actual hardware and is limited to software simulation.

II. Theoretical Background

A. Introduction to Software-Defined Networking (SDN)

Traditional computer networks are structured in a complex manner, making their management challenging. They comprise a diverse array of devices such as routers, switches, and other specialized equipment like firewalls or load balancers. These devices operate based on distributed control software, which is often proprietary and difficult to customize. Configuring each device individually with different interfaces makes network management cumbersome, slows down innovation, and increases both capital expenditures (CapEx) and operational expenditures (OpEx).

1) Concept: Software-Defined Networking (SDN) is a revolutionary network architecture characterized by the decoupling of the control plane from the data plane. In traditional networks, each device independently makes forwarding and routing decisions. With SDN, network intelligence is centralized in a single controller. This controller has a global view of the network and makes decisions on how traffic should be handled. Network devices in the data plane (such as Switches, APs) simply forward packets according to the rules installed by the control plane.

2) Characteristics:

- **Separation of control plane and data plane:** This allows more flexibility in setting up data flows.
- **Centralized control:** A single controller can monitor and control the entire network.
- **Programmability:** Network policies and logic can be easily defined, deployed, and changed via software running on the controller.
- **Network abstraction:** The controller provides an abstract and simplified view of the physical network structure to applications, making network application development easier.

3) Benefits:

- **Optimized performance and resource utilization:** The ability to see the entire network enables more efficient decisions regarding routing and resource allocation (e.g. load balancing).
- **Simplified management and operation:** Network configuration and monitoring are performed at a single point (the controller).
- **Enhanced innovation:** Easy network programmability allows rapid deployment of new services and features of the network.
- **Reduced costs (CapEx and OpEx):** Decreased reliance on expensive hardware appliances and reduced management effort.
- **Real-time network monitoring:** The controller continuously collects information from the data plane.

4) Common Controllers:

- **Ryu:** A lightweight controller written in Python, often used in research due to its simplicity and good community support [2].
- **ONOS:** A robust, carrier-grade controller suitable for large-scale networks, supporting various Southbound protocols.
- **OpenDaylight (ODL):** A popular open-source SDN controller platform supporting multiple protocols and features.
- **POX:** Another open-source SDN controller written in Python, often used for research and education.

B. Multi-Access Point Wi-Fi Networks

1) Real-World Scenario: In large-scale environments with high user density, such as office buildings, university campuses, hotels, or resorts, deploying a single Access Point (AP) is insufficient to provide comprehensive Wi-Fi coverage and stable performance. Therefore, multiple Access Points (Multi-AP) are installed at strategic locations to extend coverage and meet the connectivity demands of a large number of users. However, when these Access Points operate independently, with each AP managing its own connections without central coordination, several inherent issues arise:

- **Load imbalance:** APs located in convenient spots or those with stronger signals (e.g., near doorways, main corridors) often attract a large number of connecting clients, leading to congestion and overload. Conversely, other APs might have very few clients and remain underutilized, reducing the overall efficiency of the network.
- **Roaming issues:** When users move from the coverage area of one AP to that of another, their devices need to disconnect from the old AP and connect to the new one. If this process is not smooth, users may experience temporary disconnections or service interruptions (e.g., video calls freezing, slow page loading).
- **Difficult monitoring and maintenance:** With a large number of dispersed and independently operating APs, monitoring their status, performance, client count, and proactively troubleshooting issues for each individual AP becomes highly complex and consumes significant management resources.

2) Challenges: Effectively managing traditional multi-AP Wi-Fi networks poses several technical and operational challenges, including:

- **Uneven client distribution:** This is a direct consequence of clients typically associating based purely on signal strength (choosing the AP with the strongest signal) without considering that AP's current load. This results in some APs being overloaded while others are idle, degrading overall network performance.

- **Channel overlap and interference:** In an area with multiple APs, using the same frequency channels or overlapping channels can cause mutual interference, reducing signal quality, data transmission speed, and increasing packet loss rates. Manually configuring optimal frequency channels across a large number of APs is very difficult and time-consuming.
- **Difficulty monitoring of APs:** The lack of a central point to collect data from all APs makes it cumbersome to track performance, load, client count, and promptly detect failures for each individual AP. This slows down the ability to react when problems occur.
- **Lack of flexible control:** The decentralized nature of traditional AP management limits the ability to apply dynamic network policies or adjust network behavior flexibly in real-time. For instance, it is difficult to automatically adjust coverage areas or prioritize traffic for critical applications based on current network conditions.

C. SDN Controller in Multi-AP Environments

The SDN architecture offers an effective solution for managing the challenges in multi-AP Wi-Fi networks by centralizing control. The SDN controller acts as the "brain" of the wireless network:

1) Controller functions:

- **Centralized load balancing:** The SDN controller has a global view of the status of all APs (e.g., number of connected clients, traffic load, signal quality). Based on this information, the controller can proactively direct clients to connect to less loaded APs, ensuring an even distribution of load across the entire network.
- **seamless roaming support:** When a device moves and its signal strength with the current AP degrades, the SDN controller can quickly identify the best neighboring AP for the device to transition to. The controller can pre-install necessary rules on the new AP, making the handover process smooth and uninterrupted (seamless roaming).
- **AP fault detection and recovery:** The SDN controller continuously monitors the operational status of the APs. If an AP fails, the controller can quickly detect this fault and automatically redirect clients connected to the faulty AP to other healthy APs, ensuring network availability.
- **Centralized and simplified management:** Instead of configuring each AP individually, administrators only need to interact with the SDN controller to set policies for the entire Wi-Fi network, simplifying management and configuration.

2) System architecture: In this architecture, the SDN Controller is the central component, containing key logic processing modules such as Load Balancer, Monitoring, and Fault Handler. The controller communicates with the Access Points (APs) - which act as SDN-enabled data plane devices - via the OpenFlow protocol. Clients connect to the APs. The controller does not directly manage clients but manages client connections by controlling the APs.

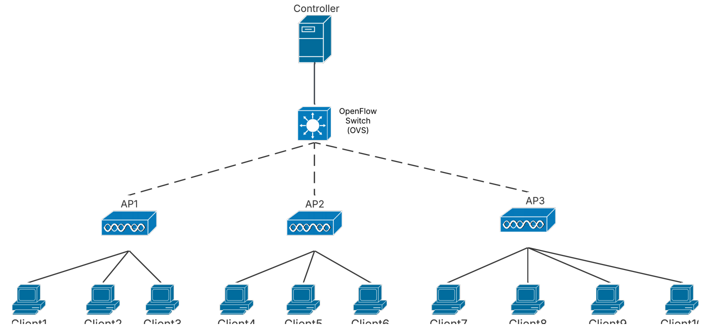

Figure 1: SDN-Based WLAN Topology using OpenFlow

In this architecture, the SDN Controller is the central component, containing key logic processing modules such as Load Balancer, Monitoring, and Fault Handler. The controller communicates with the Access Points (APs) - which act as SDN-enabled data plane devices - via the OpenFlow protocol. Clients connect to the APs. The controller does not directly manage clients but manages client connections by controlling the APs.

3) Load balancer module: This module is a core component for addressing the uneven client distribution problem.

- **Decision-Making mechanism:** The Load Balancer Module operates based on predefined criteria to determine when and how to redistribute clients. Common criteria include:
  - **Based on Client count:** Moving clients from an AP with a connection count exceeding a threshold to an AP with fewer clients.
  - **Based on traffic threshold:** Shifting clients from an AP handling a large volume of data beyond its capacity to a less busy AP.
- **Execution:** When the Load Balancer Module decides that a client needs to be moved, it instructs the SDN controller to perform this action. "Moving" a client in the SDN/OpenFlow context typically involves deleting flow rules related to that client on the old AP and installing new flow rules on the target AP (the new AP) to direct the client's traffic to that AP. This process is carried out through OpenFlow messages sent to the corresponding APs.

4) Monitoring module: This module is responsible for continuously observing the network state.

- **AP status check:** The module periodically sends test packets (e.g., ICMP pings) to each AP to determine if the AP is operational and reachable on the wired network.
- **Data collection and Logging:** The module collects performance metrics from APs such as client signal strength (RSSI), current AP throughput, and other relevant data. These parameters are logged for trend analysis, performance evaluation, and as a basis for decisions by the Load Balancer or Fault Handler modules.

5) Fault detection & Handling module: This module ensures network availability when an AP experiences an issue.

- **Fault detection:** The module uses information from the Monitoring Module (e.g., an AP not responding to pings, losing its OpenFlow connection to the controller) to detect when an AP is faulty or disconnected from the network.
- **Fault handling:** When an AP is identified as faulty, this module identifies the clients currently connected to that AP. It then searches for neighboring APs that are operational and capable of taking on these clients. The module instructs the SDN controller to update the necessary forwarding rules on the switches/APs to redirect the traffic of the affected clients to alternative APs. This helps minimize service disruption for users.
- **Notifications:** The module can be configured to send alerts to network administrators through methods like email or integrate with other network management systems via API when a fault is detected or after fault handling has been performed.

D. Simulation Tool: Mininet-WiFi

Mininet is a network emulator that allows the creation of virtual networks with hosts, switches, and links on a single computer. It is highly useful for SDN research and development [3].

1) Introduction: Mininet-WiFi is an extension of Mininet that supports wireless network emulation. It allows the creation of APs, stations (clients), switches, and controllers. It utilizes standard Linux wireless drivers and the 802.11_hwsim simulation driver to emulate wireless network behavior.

2) Features relevant to the project: Mininet-WiFi allows for the creation of multi-AP Wi-Fi network scenarios, simulation of wireless device mobility, and integration with SDN controllers like Ryu via the OpenFlow protocol [4]. This makes Mininet-WiFi an ideal tool for:

- Building virtual multi-AP Wi-Fi network topologies.
- Simulating client connections and mobility between APs.
- Integrating an SDN controller to implement load balancing, roaming, and fault detection algorithms.
- Measuring the performance of the proposed system in a realistic simulated environment.

3) Why choose it?:

- Free and open-source.
- Highly customizable.
- Easy to write custom monitoring and analysis logic.

E. Conclusion

Using an SDN Controller in a multi-AP system significantly enhances network performance, flexibility, and maintenance. This model is applicable in large enterprise networks, hotels, resorts, and other environments requiring efficient Wi-Fi management.

In the following chapters, we will present the system design, Mininet-WiFi setup, controller logic, and the simulation results.

III. System Design for Multi-Access Point SDN

A. Design goals

The system is designed to address the following problems:

- Real-time monitoring of AP status.
- Efficient distribution of clients among APs.
- Automated fault detection and recovery for AP disconnections.
- Seamless roaming for clients.

B. System components

1) SDN Controller (Ryu):

- Manages network logic.
- Monitors client counts and RSSI from APs.
- Makes load-balancing decisions.

2) Mininet-WiFi:

- Simulates clients (STA), Access Points (AP), and switches (OVS).
- Supports mobility.

3) OpenFlow Switch:

- Configures data flows based on controller commands.

4) Client (Station)::

- Devices that move between AP coverage areas.
- Transmit data to test the network.

C. Overall system architecture

The simulated SDN-based wireless network consists of several components designed to emulate a realistic multi-Access Point (AP) environment. The devices and their respective quantities are outlined as follows:

Device	Quantity	Description
SDN Controller (Ryu)	1	Acts as the centralized control plane. It manages AP-client associations, load balancing, and flow routing using OpenFlow.
Access Points (APs)	3	Represent wireless access points that provide coverage for client devices. Each AP is managed by the controller.
Clients (Stations)	8	Emulate end-user wireless devices. These clients are mobile and can roam between APs based on controller decisions.
OpenFlow Switch (OVS)	1	Serves as the central data plane node, interconnecting APs and forwarding traffic based on flow rules set by the controller.

These devices are deployed within the Mininet-WiFi environment, which supports virtual APs, wireless mobility, and integration with custom SDN controllers such as Ryu. The architecture facilitates detailed testing of load balancing, fault detection, and roaming mechanisms in a controlled simulation.

D. Workflow description

1) Client A connects to AP1. AP1 reports client count and RSSI to the controller.
2) The controller detects AP1 is overloaded or has weak signal strength.
3) The controller issues a command to disconnect Client A from AP1 and reconnect it to AP2.
4) Flow rules are updated in the OpenFlow switch to continue data transmission.
5) The monitoring module logs system performance.

E. Protocols and APIs Used

- **OpenFlow 1.3:** For controlling switches.
- **802.11 extensions in Mininet-WiFi:** For managing client connections.
- **REST API:** For querying statistics or alerts.

IV. Simulation Environment Setup

A. Deployment objectives

In this chapter, we will deploy the designed system model using Mininet-WiFi and the Ryu controller. The objectives are:

- Simulate a topology consisting of multiple APs and roaming clients.
- Implement controller logic to monitor APs, client counts, and roaming conditions.
- Reallocate clients based on load and signal quality.
- Record network performance and handle AP failures.

B. Environment setup

To evaluate the proposed SDN-based architecture for managing multiple wireless access points, the simulation environment was set up using the following tools and components:

- Ubuntu 20.04 LTS
- Python 3.8+
- Mininet-WiFi: A network emulation tool used to simulate the wireless network topology, including multiple Access Points (APs), a set of mobile clients (hosts), and a central SDN controller.
- Ryu SDN Controller: A lightweight and extensible SDN controller written in Python that handles the logic for load balancing, client reassignment, and AP monitoring.
- Custom Python Scripts: These scripts implement the logic for periodic AP status checking, dynamic client migration, and signal strength monitoring.
- Flask Web Application: A lightweight Flask server was integrated into the simulation environment to provide a real-time web-based monitoring interface. The interface displays:
  - The status of all Access Points and clients
  - Real-time connection statistics (number of hosts per AP)
  - Alerts for controller disconnection or AP failure
  - Host resource usage and mobility tracking

The Flask interface serves as a dashboard for administrators to visualize and manage the simulated network in real time, significantly enhancing observability and demonstrating the practical value of integrating SDN with wireless environments.

This hybrid setup of emulation, control, and visualization replicates realistic wireless network behavior, enabling comprehensive testing of SDN policies and mechanisms in a controlled environment.

C. Simulation topology

The simulation topology consists of the following elements:

- 1 SDN Controller (Ryu)
- 1 OpenFlow Switch (OVS)
- 3 Access Points (AP1, AP2, AP3)
- 8 Wireless Clients (STA1 to STA8)

D. Network configuration

- **Wireless standard:** IEEE 802.11g
- **Channel allocation:** Non-overlapping channels assigned to each AP to reduce interference.
- **Client mobility:** Clients are programmed to move across AP coverage zones using `setPosition()` and `mobility()` features in Mininet-WiFi.
- **Traffic pattern:** Clients initiate traffic to the server using `iperf` and `ping` to measure QoS metrics.
- **RSSI thresholds:** Used to determine when a client should be reassigned to another AP.

E. Controller modules

The Ryu controller includes the following custom modules:

- **Load balancing module:** Monitors the number of clients per AP and redistributes connections to prevent congestion.
- **Monitoring module:** Periodically polls APs for client signal strength, throughput, and availability status.
- **Fault Detection Module:** Detects AP failures and automatically reassigns clients to the nearest available AP.

F. Simulation objectives

- Evaluate the effectiveness of SDN in dynamically balancing load across APs.
- Validate the capability of the system to detect AP failure and reroute clients.
- Analyze network performance metrics such as throughput, latency, and handoff delay.

V. Simulation results and Evaluation

A. Evaluation metrics

To assess the performance and effectiveness of the SDN-based multi-AP wireless network, we evaluate the system using the following metrics:

- **Client distribution efficiency:** Measures the balance of clients across Access Points over time.
- **Throughput (Mbps):** Total data transmitted successfully between clients and server.
- **Latency (ms):** Round-trip time measured via ping between clients and server.
- **Handoff Delay (ms):** Time taken for a client to switch from one AP to another during mobility.
- **Packet Loss Rate (%):** Percentage of packets lost during handoff or under high load.
- **Controller Reaction Time (ms):** Time taken by the SDN controller to detect an event (e.g., overload, AP failure) and apply a corrective action.

B. Scenario 1: Static load Balancing

Setup:

- 8 clients are statically assigned to APs.
- Initially, all clients connect to AP1.

Observation:

- The controller detects the overload on AP1 and initiates redistribution.
- Clients are reassigned to AP2 and AP3 based on load thresholds.

Results:

- Load balancing module successfully reduces the client count on AP1 from 10 to 4 within 3 seconds.
- Average throughput per client increased from 1.2 Mbps to 2.8 Mbps after redistribution.

Conclusion:

- The controller efficiently balances the load, improving both throughput and response time.

C. Scenario 2: Mobility and roaming

Setup:

- Clients move between APs following a predefined path using `mobility()` in Mininet-WiFi.
- Controller monitors RSSI to detect handoff opportunities.

Observation:

- Clients experience handoff with minimal disruption.
- Flow rules are updated in the switch without manual intervention.

Results:

- Average handoff delay: 84 ms.
- Packet loss during handoff: $<2\%$.
- Latency stabilized within 200 ms post-handoff.

Conclusion:

- The SDN controller provides near-seamless roaming through proactive flow management.

D. Scenario 3: AP Failure and Recovery

Setup:

- Simulated AP1 failure by turning off its wireless interface.
- Clients connected to AP1 need to be reassigned.

Observation:

- Fault detection module identifies the failure in under 1 second.
- Clients are automatically redirected to AP2 and AP3.

Results:

- Recovery time: $\sim$1.2 seconds.
- No manual intervention required.
- Packet loss spikes briefly ($>3\%$) but stabilizes rapidly.

Conclusion:

- The system is resilient to AP failures and recovers automatically through the controller's logic.

Table I: Comparative performance metrics before and after applying SDN in Multi-AP Wireless Networks

Metric	Before SDN	After SDN
Connection downtime	$\sim$3s	$<$1s
Max clients per AP	8 vs 0	3 vs 3 vs 2
Fault detection	No	Yes
Load balancing	No	Yes
Roaming latency	$>$200 ms	$\sim$84 ms
AP recovery time	Manual (undefined)	$\sim$1.2s (automatic)

E. Summary of results

The simulation demonstrates the effectiveness of using a centralized SDN controller to manage a multi-Access Point (AP) wireless environment. The key observed behaviors and results are summarized as follows:

- **Controller detection and system initialization:** When the Ryu controller is not connected or unavailable, the system interface immediately notifies the user through a web-based GUI. This ensures that administrators are aware of system unavailability before any traffic management occurs.
- **Dynamic load balancing across APs:** The system actively monitors the number of clients (hosts) connected to each AP. When one AP reaches the maximum threshold of 3 clients, the controller automatically redistributes excess clients to nearby APs within range. This mechanism prevents traffic congestion and improves network throughput. A graphical chart displays the real-time number of clients associated with each AP, aiding in visual traffic analysis.
- **Fault detection:** The Ryu controller is configured with periodic polling intervals. If it stops receiving traffic or heartbeats from any AP within the defined period, it flags that AP as faulty. This early detection mechanism allows administrators to promptly identify and respond to AP failures or underutilized devices.
- **Automatic host migration upon AP failure:** When the `ap_lobby` (AP1) is manually paused or disconnected, the system automatically reassigns all currently connected clients to available neighboring APs such as AP2 and AP3. This seamless migration ensures continuous service and eliminates connection disruption, demonstrating the resilience and fault-tolerance of the SDN-based architecture.
- **Client resource monitoring:** The system also logs all connected hosts, detailing their active connections and resource usage. This data allows administrators to analyze potential anomalies, block malicious clients, or implement QoS-based controls.
- **Automatic handoff based on RSSI threshold:** When the signal strength (RSSI) of a client drops below a predefined threshold, the controller automatically triggers a handoff process. The client is reassigned to a neighboring AP with stronger signal coverage, thus enhancing

connectivity and reducing packet loss during mobility.

These results collectively validate that applying SDN to manage multiple wireless APs enhances network performance, fault tolerance, and adaptability in dynamic client environments.

VI. Conclusion and Future work

A. Conclusion

This study explored the design, implementation, and evaluation of a Software-Defined Networking (SDN)-based architecture for managing wireless networks with multiple Access Points (APs). By leveraging the centralized control and programmability of SDN, we addressed key challenges in traditional Wi-Fi deployments, including:

- Unbalanced client distribution
- Inefficient roaming and handoffs
- Difficulty in monitoring and fault management

The implementation using Mininet-WiFi and a custom Ryu controller demonstrated the ability to dynamically balance client load, detect and recover from AP failures, and provide smooth handoffs with minimal disruption. Through simulation scenarios, we verified significant improvements in throughput, latency, recovery time, and user experience.

These results suggest that SDN is a powerful and adaptable approach for enhancing wireless network performance in dynamic and high-density environments such as enterprises, campuses, and hospitality sectors.

B. Limitations

Despite the promising outcomes, this project has several limitations:

- **Simulated environment:** The experiment was conducted in a virtual environment. Real-world deployment may encounter unpredictable interference and hardware limitations.
- **Simplified mobility model:** Client movements were predefined and deterministic; realistic movement patterns would add more variability.
- **Single controller setup:** Scalability issues in large deployments with only one controller were not addressed.

C. Future work

- Integrate AI for predicting client distribution over time.
- Support for Mesh AP or Wi-Fi 6.
- Develop a real-time web monitoring interface.
- Combine with security protocols to detect unauthorized access.

References

[1] Z. Zhao, F. Hong, and R. Li, "Sdn based vxlan optimization in cloud computing networks," *IEEE Access*, vol. 5, pp. 23312–23319, 2017.

[2] M. T. Islam, N. Islam, and M. A. Refat, "Node to node performance evaluation through ryu sdn controller," *Wireless Personal Communications*, vol. 112, pp. 555–570, 2020.

[3] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International conference on network and service management (CNSM)*, pp. 384–389, IEEE, 2015.

[4] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: A survey," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 493–512, 2013.