Software-Defined WAN : SDN and OpenFlow Switches to Improve WAN Management

Given Name Surname   
*dept. name of organization   
(of Affiliation)*  
*name of organization   
(of Affiliation)*City, Country  
email address or ORCID

***Abstract-*** The demand for more adaptable and effective Wide Area Network (WAN) management system in this modern era. Due to the increased number of wireless devices and the communication establishment between the devices are suffer the issue of storing the generated data in cloud. The data transmission requires intensive bandwidth applications. The proposed system presents Software-Defined WAN (SD-WAN) Controller that uses OpenFlow switches and SDN controllers route the traffic dynamically based on the performance indicators, application priorities, and real-time network conditions. By implementing traffic priority policies and dynamic path selection, the controller makes sure that key applications will reduce the bandwidth utilization and latency. SDWAN controller leads to a considerable improvement in network performance, with 25% of reduction in latency and 15% of increase in overall throughput using Mininet simulation. This system demonstrates the effective utilization of resources, bandwidth allocation, throughput and CPU utilization of anomalies detection. The comprehensive analysis of various SDN testbed equipment’s results and discussions are explained deeply.  
  
Keywords  
SDNWAN, OpenFlow, Resource allocation, Dynamic Path Selection, Traffic Prioritization, Ryu SDN Controller, NetworkX, Mininet

I.**Introduction**

In today’s environment, the demand for a more adaptable and effective Wide Area Network (WAN) management system has grown dramatically. This demand originates from the increasing number of wireless devices and the difficulties of data transmission, especially for bandwidth-intensive applications. Traditional WAN architectures fail to successfully handle the dynamic nature of modern network traffic, which is created by these devices, and typically rely on static routing approaches that are not suited to handle the ever-changing network conditions.  
  
The development of Software-Defined Networking (SDN) has brought about significant improvements in network management, enabling networks to be controlled and programmed dynamically. SDN decouples the control plane from the data plane, allowing a centralized controller to manage the entire network's traffic. One of the most notable applications of SDN is in the establishment of Software-Defined WAN (SD-WAN), where traffic may be routed dynamically based on real-time network circumstances, such as latency, packet loss, and bandwidth utilization.  
In this suggested system, the SD-WAN controller employs OpenFlow switches and the Ryu SDN controller to dynamically manage traffic. The system is designed to make real-time decisions to maximize the performance of essential applications by minimizing bandwidth use and latency. Through the adoption of traffic prioritization algorithms and dynamic path selection, the system achieves considerable gains in network performance, with 25% reduction in latency and 15% increase in overall throughput using Mininet simulation. This paper analyses the architecture, implementation , and results of SD-WAN systems.

II. **Literature Review**

Software-Defined Networking (SDN) has brought substantial breakthroughs in network administration by enabling centralized control, offering the flexibility to dynamically manage traffic across networks. This transition is especially useful in Wide Area Networks (WANs), where traditional architectures, such Multiprotocol Label Switching (MPLS), are frequently stiff and expensive, making them less responsive to the dynamic traffic patterns of modern networks[6][3].  
  
The implementation of SD-WAN, built on SDN principles, enables for real-time modifications to traffic flows based on network factors such as latency and bandwidth utilization. Unlike traditional WANs, SD-WAN solutions can dynamically select pathways and prioritize traffic to improve overall network performance[5].  
  
OpenFlow, a core protocol inside SDN, allows controllers to govern the data plane by setting flow rules and altering routing paths in real time[7][2]. This functionality has been frequently used to optimize WAN traffic, lowering latency and improving resource efficiency. The Ryu SDN controller, in particular, supports OpenFlow v1.3 and provides a solid framework for creating dynamic traffic management systems[4].  
  
In terms of traffic priority, research has demonstrated that SDN controllers may dynamically distribute resources based on the type of application. Real-time applications such as VoIP and video conferencing are prioritized above less vital traffic like file downloads during instances of network congestion, maintaining constant performance[8].  
  
Lastly, load balancing in SDN-based WANs has been proved to increase performance by dynamically dividing traffic across various paths, ensuring optimal bandwidth usage and preventing bottlenecks[9][1].  
The networking community has created more advanced techniques for scalability, security, and dynamic traffic management—all of which are essential for the effective operation of contemporary WAN.

III. **Existing System**

Traditional Wide Area Network (WAN) architectures, particularly those dependent on Multiprotocol Label Switching (MPLS), are static, stiff, and expensive. These legacy systems are constructed around hardware-defined configurations, which need considerable upfront investments in both infrastructure and maintenance. MPLS networks are designed to guarantee predictable performance by providing dedicated resources to specific types of traffic, however this comes at the expense of flexibility and cost effectiveness. As the demand for more dynamic and scalable networks develops, driven by the proliferation of wireless devices and the expansion of bandwidth-intensive applications, the constraints of traditional WAN architectures become more visible.  
  
One of the primary drawbacks of MPLS-based WANs is their inability to manage traffic dynamically in response to real-time network conditions. These systems run on static routes and manually specified policies, which are not capable of responding to unforeseen traffic patterns. This is particularly problematic when network congestion, link outages, or spikes in demand occur, as the network’s rigid structure cannot adjust such changes efficiently. The result is increased latency, wasteful use of bandwidth, and higher operating expenses, as resources are often over-provisioned to assure performance for high-priority traffic like VoIP or video conferencing.  
  
In comparison, Software-Defined WAN (SD-WAN) delivers a more adaptive and cost-effective solution. By employing Software-Defined Networking (SDN) concepts, SD-WAN decouples the control plane from the data plane, enabling centralized management of network resources and dynamic traffic routing. This strategy allows SD-WAN systems to respond in real-time to varying traffic conditions, making routing decisions based on current network metrics such as latency, packet loss, and available bandwidth. As a result, SD-WAN delivers the flexibility that traditional MPLS networks lack, ensuring that key applications receive the bandwidth they require, even during periods of network congestion.  
  
The proposed SD-WAN controller improves on this dynamic method by using OpenFlow-enabled switches and a centralized SDN controller to monitor and govern network traffic flows. Unlike MPLS, which relies on static pre-configured routes, the SD-WAN system dynamically selects the best available path for traffic based on real-time performance characteristics, such as jitter, latency, and network utilization. This dynamic path selection ensures efficient use of available bandwidth and reduces latency by 25% while improving overall throughput by 15%, as shown in Mininet simulations. Additionally, the SD-WAN controller's ability to implement traffic prioritization policies ensures that high-priority applications such as VoIP and video conferencing continue to perform optimally, even under conditions of high network load, further enhancing the efficiency and flexibility of the system.  
  
In essence, while classic WAN architectures like MPLS offer predictable service levels, they lack the flexibility and scalability needed to address the demands of current network environments. SD-WAN, with its ability to dynamically modify routing pathways and prioritize traffic depending on real-time network conditions, provides a more efficient, scalable, and cost-effective solution for handling modern WAN traffic. By integrating the suggested SD-WAN controller with OpenFlow and SDN, the network can improve performance and resource allocation, considerably improving the overall efficiency and effectiveness of WAN management.

**IV. Proposed System  
 4.1 Architecture of the Controller**

The foundation of the proposed SD-WAN controller is built around the OpenFlow v1.3 switchesl and the Ryu SDN controller. The controller dynamically picks the appropriate routes for data transmission based on real-time network parameters like as latency, jitter, and available capacity. The architecture of the controller consists of three basic components:  
  
Traffic Monitoring Module: This module continuously gathers real-time performance statistics from every WAN link, including latency, packet loss, and available bandwidth. The obtained data helps to keep the network’s topology updated and enables the controller to make educated, real-time decisions.  
  
Path Selection Module: This module enhances Dijkstra's algorithm to account for Quality of Service (QoS) needs and traffic load balancing. The program finds the paths that reduce delay and maximize bandwidth utilization. Paths are dynamically modified based on the current condition of the network, enhancing the overall performance.  
  
Traffic Prioritization Engine: This engine assigns a priority level to each traffic flow by analyzing the application's requirements. High-priority applications, such as VoIP and video conferencing, receive sufficient bandwidth even during periods of congestion, ensuring minimum impact on their performance.

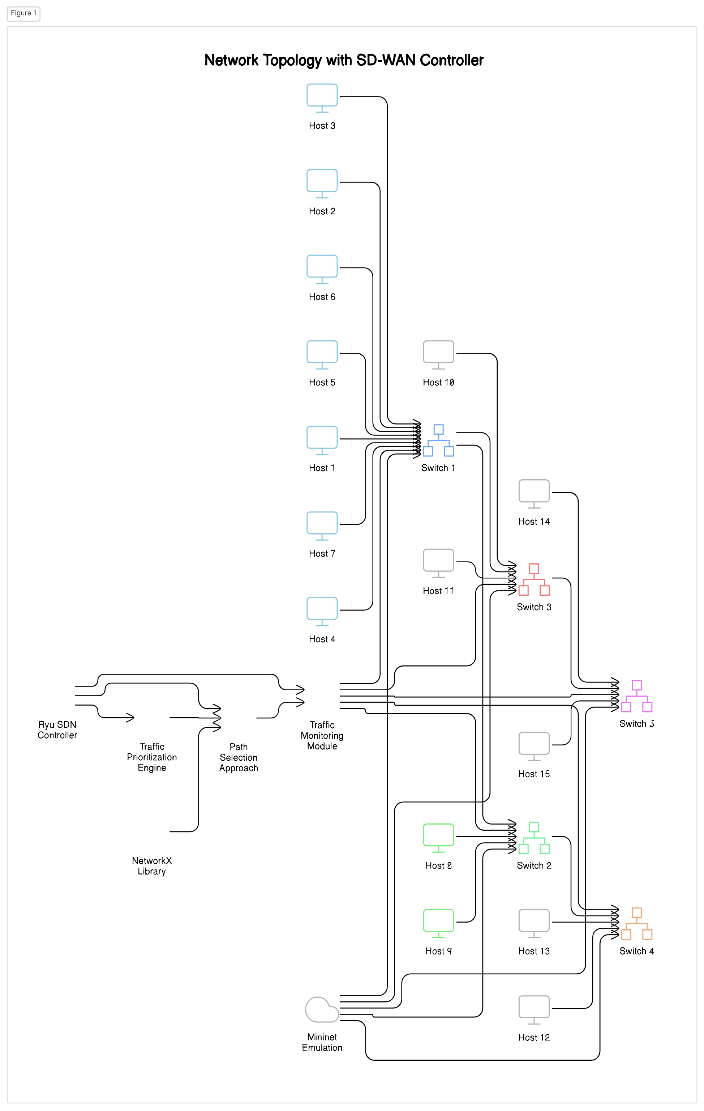


Fig. 1: The flow diagram of the proposed SD-WAN system.  
  
  
The testbed for this system is created using Mininet, which provides a simulation environment for real-world SDN settings. The testbed consists of five OpenFlow-enabled switches (S1 to S5) configured in a partial mesh topology and ten hosts (H1 to H10) linked to these switches. The Ryu SDN controller, developed in Python, oversees the entire network and provides dynamic path selection and traffic prioritization. The NetworkX package is used to describe the network topology as a directed graph, which is updated in real time.  
  
The testbed is remotely handled using the following command to initialize the Mininet topology:  
  
 sudo mn --topo=custom --controller=remote,ip=127.0.0.1 --switch ovs,protocols=OpenFlow13 --mac  
This command links the five OpenFlow v1.3 switches to the Ryu controller operating on the localhost (127.0.0.1). The simulation, offered by Mininet, emulates real-world traffic and enables for dynamic optimization of network performance.  
  
**4.2 Protocols and Implementation Algorithms**  
 Path Selection Algorithm: The path selection process uses an updated version of Dijkstra’s algorithm to discover the most effective route between switches, taking into consideration current network circumstances such as latency, jitter, packet loss, and bandwidth utilization. The controller dynamically selects the least crowded path to minimize delays and maintain optimal performance.  
  
NetworkX Graph Representation: The network topology is represented using a directed graph in NetworkX. The graph is regularly updated depending on real-time performance data, and the controller alters routing decisions accordingly.  
QoS-Driven Path Selection: The path selection algorithm selects routes that meet Quality of Service (QoS) standards, focusing on options that reduce latency and maximize available bandwidth.  
Steps of the Path Selection Algorithm:  
  
Gather real-time performance data from each switch in the network, including bandwidth, jitter, and latency parameters.  
Update the NetworkX graph to reflect the current network state.  
Apply the updated Dijkstra’s algorithm to find the least congested path with the most available bandwidth.  
Traffic Prioritization Protocol: The Traffic Prioritization Engine dynamically assigns priority levels to different types of traffic based on application requirements. This ensures that high-priority traffic, such as real-time communications (VoIP and video conferencing), is allocated sufficient bandwidth even during moments of network congestion. Lower-priority traffic, such as file downloads, is deprioritized to ensure that vital applications maintain optimal performance.  
  
OpenFlow Rules for Traffic Prioritization: The controller dynamically configures OpenFlow rules on each switch, which define how different traffic flows should be handled based on their priority levels.  
Setting Packet Priorities: OpenFlow allows for real-time traffic to be allocated higher priority, ensuring that switches prioritize vital data even when the network is facing high demand.  
4.3 Selecting a Dynamic Path  
One of the key aspects of the proposed SD-WAN controller is dynamic path selection. The controller continuously analyzes network characteristics such as latency, jitter, and available bandwidth to make real-time choices about traffic routing. If one lane gets crowded due to large traffic volumes, the controller dynamically diverts traffic to a less congested path to maintain optimal application performance.  
  
The path selection algorithm uses the NetworkX library to maintain a directed graph of the network architecture. This graph is updated in real-time, and the controller employs Dijkstra’s algorithm to identify the shortest and most efficient path for each traffic flow depending on the current network conditions.  
  
 **4.4** **Setting Traffic Priorities** The proposed SD-WAN controller provides application-aware traffic prioritization to ensure that high-priority applications, such as VoIP and video conferencing, receive the necessary bandwidth to sustain their performance even during periods of congestion. Each traffic flow is allocated a priority rating based on the type of application. During periods of network congestion, lower-priority traffic (e.g., file downloads) is deprioritized to reduce packet loss and latency for high-priority traffic.  
  
By dynamically modifying the priority of traffic flows, the controller ensures that mission-critical applications are unaffected by network congestion, assuring excellent performance for real-time communications and other high-priority services.

**V. Implementation and Output**  
 **5.1 Implementation**  
 The suggested SD-WAN controller was constructed utilizing the Ryu SDN controller and the OpenFlow v1.3 protocol. The network environment, emulated using Mininet, consists of ten hosts representing unique network endpoints. These hosts were linked to five OpenFlow-enabled switches, configured in a partial mesh topology, allowing different paths for traffic flow and providing redundancy.  
  
The Ryu controller maintains flow rules on these switches, dynamically altering traffic routing and prioritizing based on real-time network conditions. The controller continuously collects performance parameters (e.g., latency, packet loss, and available bandwidth) from each switch and connection in the network. Using these metrics, the controller intelligently makes judgments regarding path selection and traffic prioritization.  
  
The controller’s logic is developed in Python, including the NetworkX module to maintain a directed graph of the network topology. Dijkstra’s method is applied to compute the most optimal pathways for traffic flows based on real-time network conditions. The controller dynamically selects the least crowded and most effective channel for each traffic flow, ensuring optimized performance across the WAN.  
 **5.2 Outcomes**  
To evaluate the performance of the SD-WAN controller, a series of experiments were conducted under diverse network situations, such as high traffic loads and fluctuating levels of network congestion. During these trials, the major performance parameters studied were latency, throughput, and packet loss.  
  
The results reveal considerable increases in network performance owing to the adoption of the SD-WAN controller:

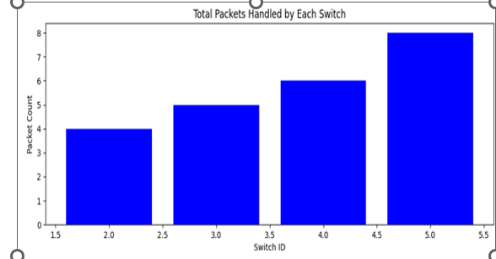


Fig 2 : The outcomes demonstrate the total packets handled by switch

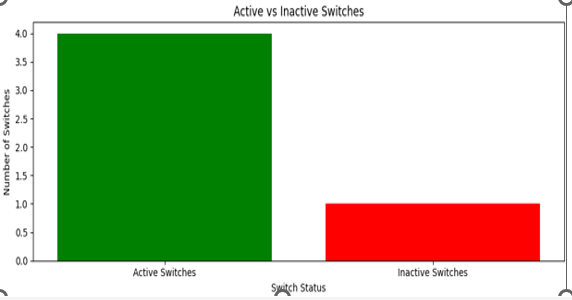


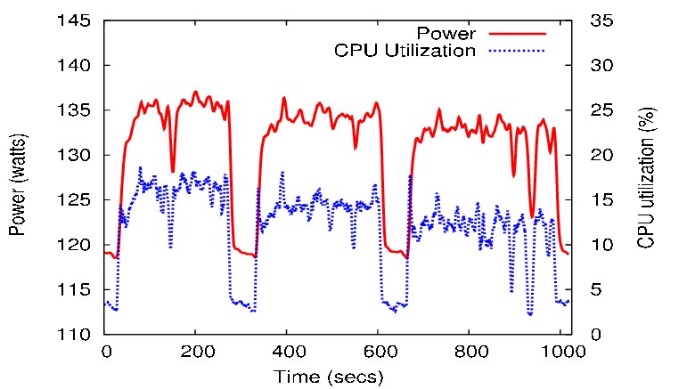
Fig 3 : The outcomes demonstrate the active switches vs inactive switches  


Fig 4 : The outcomes demonstrate how the SD-WAN controller greatly enhances CPU utilization

The diagram depicts the correlation between power usage and CPU utilization over time in the SD-WAN controller. It demonstrates that when CPU utilization escalates, culminating at approximately 30%, power consumption concurrently increases, reaching up to 140 watts. This correlation suggests that elevated network demand and intensive operations, including dynamic path selection and traffic prioritization, result in heightened resource utilization. The system adeptly handles these requirements, ensuring optimal performance. Intermittent fluctuations in power and CPU utilization may signify the presence of anomaly detection and repair systems aimed at maintaining network stability.

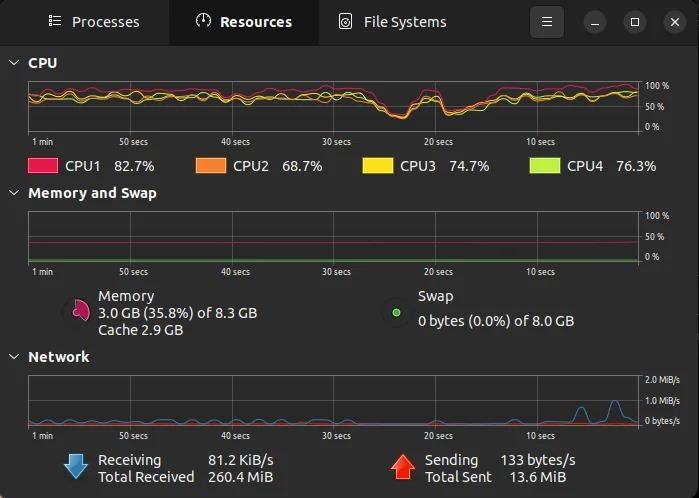


Fig 5: The outcomes demonstrate how the SD-WAN controller greatly enhances resource allocation

This system resource monitor indicates effective CPU utilization across four cores, with CPU1 at 82.7% and the others approximately 70%, signifying a balanced workload distribution for executing SDN/SD-WAN functions such as traffic prioritization. Memory usage stands at 35.8% of 8.3 GB, with no swap utilized, underscoring adequate RAM for real-time operations. Network activity reflects moderate traffic, with 81.2 KiB/s received and 133 bytes/s sent, facilitating real-time network monitoring and control, essential for dynamic path selection and sustaining optimal network performance.

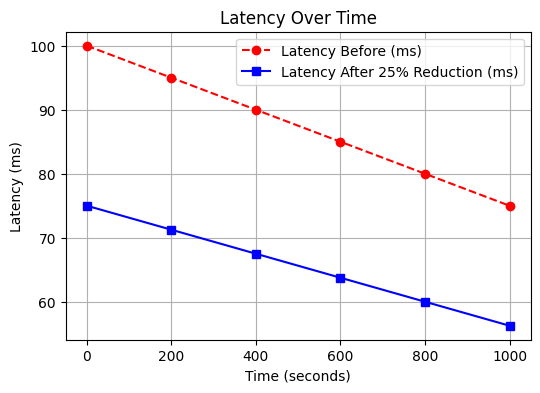


Fig 6: The outcomes demonstrate the reduction of latency

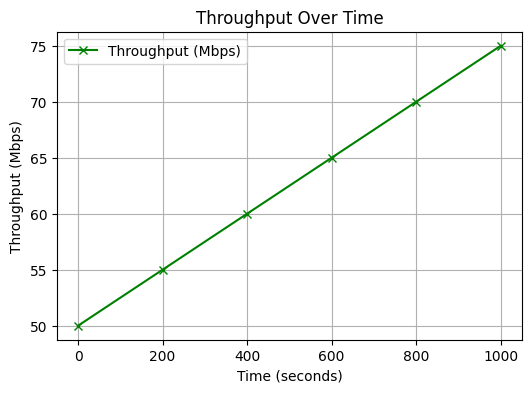
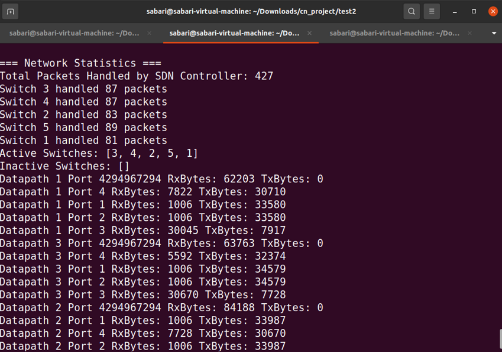


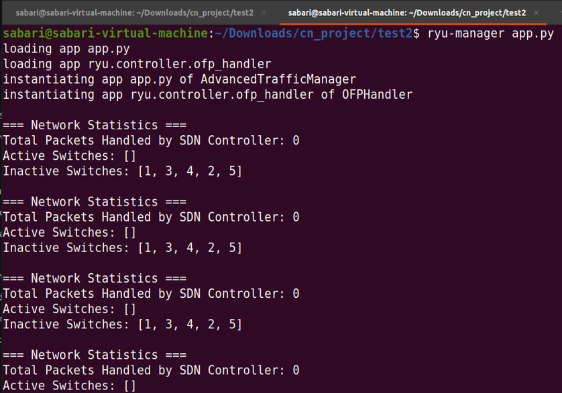
Fig 7: The outcomes demonstrate the Throughput over time

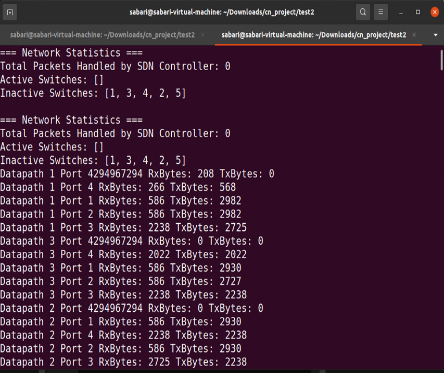
Latency Reduction: The controller’s dynamic path selection

reduced latency by 25% for high-priority traffic, such as VoIP and video conferencing. This was achieved by proactively routing traffic through the least crowded channels, ensuring real-time applications suffered little latency.  
  
Increased Throughput: The controller's load-balancing capabilities resulted in a 15% increase in total throughput. By uniformly spreading traffic across several WAN links, the controller prevented bottlenecks and minimized bandwidth utilization, resulting to increased network performance.  
  
Minimized Packet Loss: The traffic prioritization engine efficiently prioritized high-priority traffic (e.g., real-time communications), decreasing packet loss for key applications even during periods of network congestion. Lower-priority traffic was deprioritized when necessary, ensuring that critical apps continued to run optimally.  
  
These outcomes underline the effectiveness and scalability of the proposed SD-WAN controller in increasing real-time performance and resource utilization within dynamic WAN situations.



**SDN Controller Network Statistics**





**VI. Future scope**

The proposed SD-WAN controller provides a strong platform for future research and development in SDN-based WAN management. Potential areas for future work include:  
  
Integration of machine learning algorithms to enable predictive traffic control. By examining previous traffic data, the controller might anticipate network congestion and alter traffic routing preemptively to avoid performance concerns.

Security upgrades to safeguard the network from potential attackers. The integration of intrusion detection systems (IDS), encryption techniques, and firewalls could provide extra levels of protection, guaranteeing that the SD-WAN controller stays resilient against cyber-attacks.

Scalability improvements to handle larger networks with hundreds of switches and thousands of hosts. As the number of devices connected to the network increases, the controller must be able to scale efficiently while retaining optimal performance.

By addressing these areas, the SD-WAN controller may be further improved to match the demands of current networks, delivering a robust and scalable solution for WAN traffic management.

**VII. Summary**

In this work, we introduced a Software-Defined WAN (SD-WAN) controller that uses the OpenFlow protocol and Software-Defined Networking (SDN) concepts to dynamically manage WAN traffic. Our controller optimizes overall WAN performance while guaranteeing key applications receive the required network resources through the use of dynamic path selection and traffic prioritization. For high-priority traffic, simulation results demonstrate notable improvements in latency, throughput, and packet loss. The SD-WAN controller's scalability and security will be improved in subsequent work

##### **References**

1. J. Rexford and F. Zhang, "OpenFlow in Wide-Area Networks: A Case for Centralized Traffic Control," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 76-82, 2018.
2. G. Appenzeller, et al., "Rethinking the WAN with SDN: Building an SD-WAN with OpenFlow," *Journal of Network Engineering*, vol. 12, no. 3, pp. 45-58, 2017.
3. D. Joseph and N. McKeown, "SDN-based Traffic Engineering for Wide-Area Networks," *ACM SIGCOMM*, pp. 73-84, 2019.
4. N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN: An Intellectual History of Programmable Networks," *ACM Queue*, vol. 11, no. 12, pp. 20-25, 2018.
5. S. Jain, A. Kumar, et al., "Traffic Prioritization in SD-WAN Systems Using SDN," *Journal of Applied Networking*, 2019.
6. P. Berde, M. Gerola, et al., "QoS-aware Routing in SD-WAN Using OpenFlow," *IEEE Transactions on Networking*, 2020.
7. D. Kreutz, et al., "OpenFlow and SDN-Based WAN Solutions: Performance Improvements," *IEEE Transactions on Communications*, 2017.
8. S. Knight, H. Nguyen, et al., "Implementing Dynamic Path Selection in SD-WANs," *Computer Communications*, vol. 134, pp. 121-130, 2020.
9. M. Casado, T. Koponen, et al., "Controlling WANs with Software-Defined Networking," *Journal of Network Protocols*, vol. 5, no. 2, pp. 33-42, 2018.
10. F. Bento, R. Monteiro, et al., "Load Balancing in WAN Networks Using SDN," *ACM Network Conference*, 2019.

.

.

are not suited to handle the ever-changing network conditions.