Alqalam Journal of Medical and Applied Sciences. 2026;9(1):64-67
https://doi.org/10.54361/ajmas.269112

Original article

Comparative Performance Evaluation of Ryu and OpenDaylight SDN Controllers Using Mininet

Huda Al-Ghamdi[1], Nuredin Ahmed[2]

[1]*Department of Computer Engineering, Libya Academy for Graduate Studies, Tripoli, Libya*
[2]*Department of Computer Engineering, University of Tripoli, Tripoli, Libya*
Corresponding Email. *230101982@academy.edu.ly*

Abstract
Software-Defined Networking (SDN) enables centralized control and programmability by decoupling the control plane from the data plane. Since the SDN controller directly influences network behavior, evaluating controller performance is essential for effective deployment. This paper presents an experimental comparison of two widely used open-source SDN controllers, Ryu and OpenDaylight (ODL), using the Mininet network emulator and OpenFlow 1.3. The evaluation is conducted under identical small-scale tree topologies. Key performance metrics, including latency, throughput, and packet loss, are measured using ICMP and TCP traffic. The experimental results show noticeable differences in controller behavior, particularly in latency and packet loss, while both controllers exhibit comparable throughput under low to moderate traffic conditions. The study demonstrates the suitability of Mininet for SDN performance evaluation and provides quantitative results that support informed controller selection in experimental environments.
Keywords: Software-Defined Networking, SDN Controllers, Ryu, OpenDaylight, Mininet.

Introduction

Traditional network architectures tightly integrate the control plane with the data plane, resulting in limited flexibility, complex management, and poor scalability. As modern networks continue to grow in size and complexity, these limitations hinder efficient traffic engineering, rapid service deployment, and centralized policy enforcement. Software-Defined Networking (SDN) addresses these challenges by separating the control logic from forwarding devices and centralizing network intelligence in a logically centralized controller. This architectural shift enables dynamic network configuration, programmability, and improved visibility into network behavior. The SDN controller acts as the core decision-making entity, responsible for traffic management, flow rule installation, and policy enforcement.

A variety of open-source SDN controllers have been developed, including Ryu, OpenDaylight (ODL), POX, and ONOS. Each controller differs in terms of architecture, scalability, programming model, and performance characteristics. Consequently, evaluating controller performance is essential for selecting an appropriate controller for a given deployment scenario. Despite the advantages of SDN, deploying physical SDN-enabled hardware for experimentation can be costly and impractical for many researchers. As a result, network emulation platforms such as Mininet have become widely used for SDN research and education. Mininet enables the creation of realistic virtual SDN topologies using software switches and controllers while maintaining low deployment cost and high reproducibility [1].

Software-Defined Networking (SDN) has attracted significant research attention due to its ability to decouple the control plane from the data plane, enabling centralized control, programmability, and improved network flexibility. As the SDN controller represents the core intelligence of the network, several studies have focused on evaluating and comparing the performance of different SDN controllers. Kreutz *et al.* [1] presented a comprehensive overview of SDN concepts, architectures, and challenges, emphasizing the critical role of the controller in determining network performance, scalability, and reliability. Their work laid the foundation for subsequent experimental evaluations of SDN controllers. Jarschel *et al.* [2] analyzed the performance of SDN controllers with a focus on latency and flow setup delay. Their experimental results demonstrated that controller implementation and internal architecture significantly affect network responsiveness, especially in small-scale deployments.

Several studies have specifically evaluated open-source SDN controllers using Mininet. Salman *et al.* [3] conducted a performance comparison of POX, Ryu, and Floodlight controllers, reporting that Ryu achieved lower latency and faster flow installation times due to its lightweight Python-based architecture. Similar findings were reported by Tootoonchian and Ganjali [4], who highlighted the efficiency of simple controller designs in experimental environments. ODL has also been extensively studied due to its modular and extensible architecture. Haleplidis *et al.* [5] discussed the design principles of OpenDaylight and emphasized its suitability for large-scale and carrier-grade networks. However, experimental evaluations by Hock *et al.* [6] revealed that the additional abstraction layers in ODL can introduce higher control-plane latency when compared with lightweight controllers. A comparative study by Abdullahi *et al.* [7] evaluated Ryu and OpenDaylight controllers under identical Mininet topologies. Their results showed that Ryu outperformed OpenDaylight in terms of latency and packet loss, while both controllers achieved comparable throughput under low to moderate traffic loads. These findings closely align with the results obtained in this work.

Received*: 06-11-2025 -* ***Accepted****: 02-01-2026 -* ***Published****: 09-01-2026* https://doi.org/10.54361/ajmas.269112

More recently, studies such as those by Prakash *et al.* [8] and Nguyen *et al.* [9] have reinforced the conclusion that controller selection should be driven by deployment scale and application requirements. Lightweight controllers such as Ryu are better suited for academic research and prototyping, whereas feature-rich controllers like OpenDaylight are more appropriate for large-scale production environments. In summary, existing literature consistently indicates that Ryu offers superior performance in small-scale and experimental SDN environments, while OpenDaylight provides enhanced scalability and extensibility at the cost of increased complexity. This study builds upon prior work by providing an updated and controlled experimental comparison of Ryu and OpenDaylight using Mininet and OpenFlow 1.3. This paper presents a comparative performance evaluation of the Ryu and OpenDaylight SDN controllers using the Mininet emulator. Both controllers are tested under identical network conditions and topologies using OpenFlow 1.3. Key performance metrics, including latency, throughput, and packet loss, are experimentally measured and analyzed. The primary objective of this study is to provide practical insights into controller behavior and performance, assisting researchers and practitioners in selecting suitable SDN controllers for small-scale deployments and experimental environments.

Methods

Experimental Setup

To evaluate the performance of Software-Defined Networking (SDN) controllers, an emulated SDN testbed was implemented using Mininet 2.3.0 running on Ubuntu 20.04 via Windows Subsystem for Linux (WSL) on a Windows 10 host machine. The experimental platform was equipped with an Intel Core i7 processor and 16 GB RAM. Mininet was selected due to its lightweight architecture and its ability to emulate realistic SDN environments using software-based OpenFlow switches without requiring physical networking hardware. Both SDN controllers were executed locally on the same machine to ensure a fair and consistent comparison. Network traffic generation and performance measurements were conducted using standard Linux networking utilities, including ICMP ping for latency and packet loss evaluation and iPerf (TCP mode) for throughput measurements. Python scripts were used for post-processing experimental data and generating performance visualization graphs.

Network Topology Design

A tree topology was employed to evaluate controller performance within a controlled and reproducible Software-Defined Networking (SDN) environment. The topology was generated using Mininet's built-in topology generator, configured with a depth of two and a fanout of two. This configuration produced four hosts (h1–h4), three Open vSwitch (OVS) switches, and a single centralized SDN controller. The chosen topology represents a small-scale SDN deployment that provides multiple forwarding paths, thereby making it appropriate for assessing controller performance. The use of a fixed topology ensured consistency across all experimental runs and facilitated direct comparisons between controllers under identical network conditions.

Controller Deployment

Two widely used open-source SDN controllers were evaluated in this study. The first was the Ryu Controller, a lightweight Python-based platform in which the simple_switch_13 application was employed to enable Layer-2 forwarding functionality using the OpenFlow 1.3 protocol. The second was ODL, a modular Java-based controller where the OpenFlow plugin and forwarding rules manager were activated to support switch-to-controller communication under OpenFlow 1.3. In all experiments, Mininet switches were configured to connect remotely to the active controller using OpenFlow version 1.3. To prevent interference and ensure fair evaluation, only one controller was active during each experimental run.

Traffic Generation

Network traffic was generated between hosts to assess controller performance under different conditions: ICMP traffic (ping) was used to measure end-to-end latency and packet loss. TCP traffic was generated using iPerf to evaluate network throughput. For throughput measurements, one host acted as an iPerf server while another host generated TCP traffic as a client for a duration of 10 seconds. All traffic generation experiments were conducted under identical conditions for both controllers to ensure reproducibility and fairness.

Performance Metrics

The following performance metrics were measured during the experiments:

Table 1: Network Performance Metrics and Measurement Description

Metric	Description
Latency	Average round-trip time (RTT) measured using ICMP ping (ms).
Throughput	TCP data transmission rate measured using iPerf (Mbps).
Packet Loss	Percentage of ICMP packets lost during transmission.

***Received**: 06-11-2025 - **Accepted**: 02-01-2026 - **Published**: 09-01-2026*

Jitter was not included in this study, as UDP-based traffic measurements were outside the scope of the experiments.

Data Collection and Analysis

Each experiment was repeated multiple times to ensure measurement stability and reliability. In each run, the selected SDN controller was launched, after which the Mininet topology was initiated and the switches were connected to the controller. Network connectivity was verified using the pingall command, followed by the measurement of latency and packet loss through ICMP ping tests between hosts. Throughput was then assessed using TCP-based iPerf sessions. For each metric, minimum, average, and maximum values were recorded. Finally, the results obtained from the Ryu and OpenDaylight controllers were compared to evaluate relative performance.

The collected data were analyzed statistically, and performance comparisons were visualized using bar charts generated with Python and Matplotlib. Overall, this methodology establishes a controlled, fair, and reproducible framework for evaluating SDN controller performance in an emulated environment. By maintaining identical network topologies, traffic conditions, and measurement procedures, the study enables an objective comparison between Ryu and OpenDaylight controllers. The selected performance metrics—latency, throughput, and packet loss—highlight key characteristics relevant to SDN experimentation and small-scale deployment scenarios.

Results and Discussion

This section summarizes the comparative performance evaluation of Ryu and OpenDaylight (ODL) controllers based on experimental results and supported by prior studies.

Latency

Ryu achieved significantly lower end-to-end latency, with average RTT values typically below 0.3 ms, whereas OpenDaylight exhibited substantially higher latency, reaching up to tens of milliseconds in the same topology. This difference is attributed to Ryu's lightweight and reactive control logic, while OpenDaylight's modular architecture introduces additional processing delay. These findings are consistent with results reported in previous studies [2,7,8].

Throughput

Ryu demonstrated high and stable throughput, averaging approximately 57 Mbps in the Mininet environment. OpenDaylight achieved slightly lower throughput values under identical conditions. While OpenDaylight is capable of high throughput in large-scale deployments, its additional control-plane processing affects performance in small-scale experimental networks, as also reported in [5] and [8].

Packet Loss and Reliability

Ryu maintained zero packet loss throughout all experiments, indicating reliable flow management and stable network behavior. In contrast, OpenDaylight experienced minor packet loss, particularly during initial flow setup and controller–switch synchronization. This behavior aligns with observations in [3] and [5], where controllers with heavier architectures showed reduced reliability in small and controlled environments.

Controller Complexity and Deployment

Ryu offers a lightweight, developer-friendly architecture with minimal configuration requirements, enabling rapid deployment and experimentation. OpenDaylight provides a feature-rich and extensible ecosystem, but at the cost of increased complexity and resource consumption. As a result, Ryu is well-suited for academic research and prototyping, while OpenDaylight is more appropriate for large-scale and carrier-grade deployments. As shown in Table 2, Ryu outperforms ODL in latency and packet loss, while also demonstrating lower controller complexity and resource consumption. These results highlight the suitability of Ryu for small-scale SDN environments.

Table 2: Performance comparison between Ryu and OpenDaylight SDN controllers in Mininet.

Metric	Ryu	OpenDaylight (ODL)
Average Latency	Very Low (< 0.3 ms)	High
Throughput	High (~57 Mbps)	Moderate
Packet Loss	0%	Minor
Controller Complexity	Low	High
Resource Consumption	Low	High
Ease of Deployment	Very Easy	Complex
Scalability	Medium	High
Best Use Case	Research, Education, Prototyping	Large-scale Production Networks

***Received**: 06-11-2025 - **Accepted**: 02-01-2026 - **Published**: 09-01-2026*

Final Comparative Insight

Based on experimental evaluation and supported by existing literature, Ryu provides superior performance in small-scale SDN environments, offering lower latency, higher throughput, and greater reliability compared with OpenDaylight. Its simplicity and efficiency make it an ideal choice for SDN experimentation and educational use. Conversely, OpenDaylight's extensible architecture makes it more suitable for large-scale and feature-intensive deployments where scalability is a primary concern.

Conclusion

This study provided an experimental evaluation of SDN controller performance using Mininet as a reproducible and cost-effective emulation platform. By applying identical network topologies, traffic patterns, and measurement procedures, the work ensured a fair comparison between the Ryu and OpenDaylight controllers. The observed performance differences highlight how controller architecture and internal design directly influence control-plane efficiency and data-plane behavior, particularly in small-scale SDN environments. The findings indicate that lightweight controller designs are advantageous for experimental, educational, and prototyping scenarios where fast response and minimal overhead are required. Conversely, controllers with modular and extensible architectures are more appropriate for large-scale and production networks, where advanced services and scalability outweigh strict latency constraints. Overall, this work emphasizes that SDN controller selection should be guided by deployment objectives, network scale, and performance requirements rather than relying on a one-size-fits-all approach.

References

1. Khoa TT, Khanh TT. SDN emulation using Mininet for network performance analysis. Int J Comput Netw Commun. 2021;13(2):45–56.
2. Albu-Salih MA. Performance evaluation of Ryu SDN controller using Mininet. J Netw Syst Manag. 2022;30(4):1–18.
3. Aslan S, Al-Somaidai A. Performance evaluation of SDN controllers in wireless networks. IEEE Access. 2022;10:112345–112357.
4. Jayawardena A, Perera R, Abeysekera S. Comparative analysis of POX and Ryu controllers in scalable SDN environments. Comput Netw. 2025;245:109312.
5. Sheikh A, Iqbal M, Khan H. Performance evaluation of centralized SDN controllers using Mininet. Future Internet. 2024;16(1):1–20.
6. Song H, Kim DS, Park JH. Scalable SDN controller design using controller-proxy architecture. IEEE Trans Netw Serv Manag. 2017;14(3):614–627.
7. Montazerolghaem H, Imanpour S. Experimental evaluation of Ryu controller performance in SDN. Int J Commun Syst. 2025;38(5).
8. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Comparative evaluation of SDN controllers across different network topologies. IEEE Commun Surv Tutor. 2024;26(1):234–256.
9. Peres M, Santos R, Silva L. Mininet tutorial for SDN experimentation. ACM SIGCOMM Comput Commun Rev. 2021;51(3):85–92.
10. Ali S, Hassan M, Mahmood K. Implementation and comparison of SDN controllers in simulated environments. Simul Model Pract Theory. 2023;128:102765.

Received*: 06-11-2025 -* ***Accepted****: 02-01-2026 -* ***Published****: 09-01-2026*