INFOTECH journal
ISSN : 2460-1861 (Print), 2615-4250 (Online)
Vol. 11 No. 2 Desember 2025, pp. 408-415

PERFORMANCE ANALYSIS OF THE OPENDAYLIGHT (ODL) SDN CONTROLLER ON TREE TOPOLOGY: THROUGHPUT, JITTER, AND PACKET LOSS

Asruddin Asruddin[1], Ade Davy Wiranata[2*]
[1]Fakultas Ilmu Komputer, Program Studi Sistem Komputer, Universitas Bung Karno, Jakarta, Indonesia
[2]Fakultas Teknologi Industri dan Informatika, Studi Teknik Informatika, Universitas Muhammadiyah Prof. Dr. HAMKA, Jakarta, Indonesia
Responden: adedavy@uhamka.ac.id

ABSTRACT

Software Defined Networking (SDN) separates the control plane from the data plane and enables centralized, programmable network management. This study evaluates the Quality of Service (QoS) performance of the OpenDaylight (ODL) controller on a tree (3,2) topology emulated using Mininet. The experimental setup enforces per-link shaping at 10 Mbps bandwidth and 2 ms delay across seven Open vSwitch nodes. Unlike prior studies that report aggregated QoS values, this work (i) standardizes receiver-side QoS metrics for TCP/UDP throughput, jitter, and packet loss, and (ii) provides per-host QoS measurements together with averages and standard deviations to transparently capture performance distribution across the topology. Using iperf3-generated flows from seven senders to a single receiver, the results show average UDP throughput of 8.79 Mbit/s, TCP throughput of 8.42 Mbit/s, average jitter of 4.12 ms, and negligible packet loss (0.086%). These findings demonstrate that OpenDaylight, combined with deterministic link shaping, provides stable QoS performance in tree-based SDN environments. The proposed measurement methodology may serve as a reproducible baseline for follow-up evaluations involving different controllers or queueing disciplines.

***Keywords**: Software Defined Networking, OpenDaylight, Tree Topology, QoS, Throughput, Jitter, Packet Loss..*

Article History :
Received date : 19-10-2025
Revised date : 08-11-2025
Published date : 01-12-2025

DOI :
https://doi.org/10.31949/infotech.v11i2.16053

1. INTRODUCTION

The rapid advancement of networked systems has significantly increased the volume, density, and complexity of data traffic in modern communication infrastructures. These demands challenge traditional network architectures, which tightly couple the control plane and data plane, resulting in rigid, static, and difficult-to-manage environments. As traffic patterns evolve dynamically, conventional architectures struggle to deliver stable service availability and consistent communication quality, especially under dense multi-hop topologies (Hamad, Yalda, & Tapus, 2023).

Software Defined Networking (SDN) emerges as a programmable networking paradigm that decouples the control plane from the data plane, enabling centralized, flexible, and policy-driven network management. Through OpenFlow-based control, SDN controllers can implement traffic engineering, bandwidth allocation, routing decisions, and QoS policies more efficiently. Among open-source controllers, OpenDaylight (ODL) is widely used for academic and experimental scenarios because it supports modular components, OpenFlow 1.3 compatibility, and topology visualization (Kaczmarek & Litka, 2024).

Quality of Service (QoS) evaluation plays a crucial role in assessing SDN performance, particularly in measuring throughput, jitter, packet loss, and bandwidth utilization. Several studies have investigated SDN controller performance across different topologies. (Hui, Hoh, Ong, Zhu, & Yoon, 2025) demonstrated that controller type and topology shape influence latency and throughput characteristics. Bandwidth management studies (Putra & Suartana, 2022; Tahir, Sugara, & Harsiwi, 2022) showed that queueing mechanisms such as HTB on Open vSwitch (OVS) improve throughput stability under controlled scenarios. (Caroline, Basuki, & Dewanta, 2022) examined network slicing performance on Mininet with POX and found that the number of switches and topology design substantially impact packet delivery performance. Another study on OpenDaylight with a tree topology (Eka Pratama & Bakkara, 2021) identified performance variation but lacked three key aspects: explicit per-link capacity shaping, consistent receiver-side goodput metrics, and transparent per-host performance reporting.

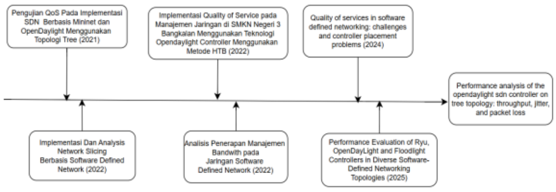
From these findings, three research gaps are identified:

a. The absence of realistic QoS testing with deterministic per-link shaping (bandwidth and delay);
b. Inconsistent definitions of receiver-side throughput and packet loss across TCP and UDP flows; and
c. Limited transparency in reporting per-host QoS distribution, particularly in hierarchical topologies such as tree (3,2).

This study aims to address these gaps by evaluating the QoS performance of the OpenDaylight SDN controller on a tree (3,2) topology using Mininet with explicit shaping of 10 Mbps bandwidth and 2 ms delay on every link. The analysis focuses on standardized receiver-side metrics, including TCP/UDP throughput, jitter, packet loss, and bandwidth utilization. The experimental design is fully reproducible and reports per-host results supported by mean and standard deviation to provide a holistic understanding of performance distribution.

The contributions of this paper are as follows:

a. A standardized receiver-side QoS metric definition for both TCP and UDP throughput and packet loss, ensuring consistent and comparable measurements.
b. A transparent per-host QoS reporting methodology that includes individual results, mean, and standard deviation across all branches of the tree topology.
c. A reproducible experimental framework for SDN QoS evaluation using deterministic per-link bandwidth and delay shaping, serving as a baseline for future controller comparison studies.



Gambar 1. Research Roadmap

2. RELATED WORK

Research on SDN performance evaluation has expanded across several domains, including controller responsiveness, traffic engineering, topology-specific performance, and QoS measurement. Recent studies emphasize that controller behavior and flow-rule processing significantly influence QoS stability in multi-hop environments (Hamad, Yalda, & Tapus, 2023).

(Hui, Hoh, Ong, Zhu, & Yoon, 2025) conducted comparative testing across single, linear, and tree topologies and reported that hierarchical structures such as tree topologies introduce greater variability in latency and throughput due to multi-hop dependency. However, the study did not include deterministic per-link shaping, which limits replicability. (Putra & Suartana, 2022) and (Tahir, Sugara, & Harsiwi, 2022) demonstrated that queueing mechanisms (e.g., HTB) and bandwidth management on OVS improve throughput consistency, although their experiments focused on small-scale networks and did not explore hierarchical tree structures.

(Caroline, Basuki, & Dewanta, 2022) evaluated SDN-based network slicing using the POX controller and confirmed that topology size and the number of switches significantly affect end-to-end performance. Their results support the notion that topology design must be considered in QoS evaluation but lacked receiver-side measurement definitions, which may cause discrepancies between sender-reported and receiver-observed throughput.

(Eka Pratama & Bakkara, 2021) examined SDN QoS under OpenDaylight using Mininet with a tree topology and noted general performance trends. However, three limitations remain:

a. No consistent per-link bandwidth shaping;
b. No standardized receiver-side goodput metrics;
c. And no per-host QoS distribution analysis.

These limitations make it difficult to perform transparent and reproducible comparison across studies.

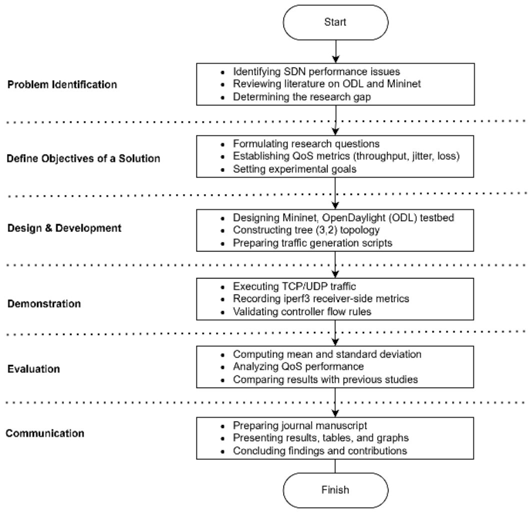
In contrast to prior works, this study implements explicit bandwidth and delay shaping on every link, uses strictly receiver-side metrics for TCP and UDP flows, and reports per-host statistics including averages and standard deviations. This approach strengthens result comparability, improves reproducibility, and directly addresses the gaps identified in previous research.

3. METHOD

This study adopts an experimental research methodology to evaluate the QoS performance of the OpenDaylight SDN controller operating on a tree (3,2) topology using the Mininet emulator. The methodology consists of five major components: (1) research framework, (2) testbed environment, (3) topology design, (4) traffic generation procedure, and (5) standardized receiver-side metric definitions. All configurations were designed to ensure deterministic, transparent, and fully reproducible experiments.

3.1. Research Framework

The research follows a structured workflow based on the Design Science Research Methodology (DSRM), covering problem identification, literature review, experimental design, implementation, evaluation, and documentation. The roadmap is shown in Figure 2.



Gambar 2. Design Science Research Methodology (DSRM)

3.2. Experimental Testbed Environment

All experiments were performed on a single-host machine running Ubuntu Linux, configured as the Mininet emulator and OpenDaylight controller server. The hardware and software specifications are listed in Table 1.

Tabel 1. Experimental Testbed Specifications

Component	**Specification**
Operating System	Ubuntu LTS (64-bit)
CPU	Quad-core 2.4–3.0 GHz
Memory	8–16 GB RAM
Storage	SSD 256–512 GB
Emulator	Mininet 2.x
Virtual Switch	Open vSwitch (OVS) 2.x
SDN Controller	OpenDaylight Carbon 0.6.1
Protocol	OpenFlow 1.3
Traffic Generator	iperf3 (TCP/UDP)

The OpenDaylight controller was installed with the OpenDaylight-L2Switch, OpenFlowPlugin, and DLUX modules. OpenDaylight was configured to listen on port 6653 for OpenFlow 1.3 connections.

3.3. Experimental Testbed Environment

A tree (3,2) topology was selected to reflect hierarchical multi-hop scenarios commonly found in enterprise and campus networks. The topology consists of:

limitations remain:

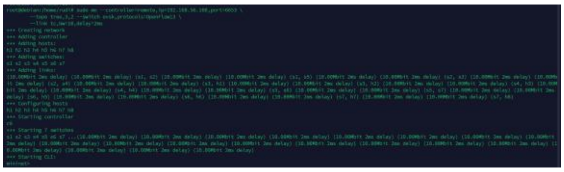
a. 7 switches (s1–s7), arranged in three layers
b. 8 hosts (h1–h8), attached to the lowest-layer switches
c. Uniform link shaping: 10 Mbps bandwidth and 2 ms delay on every link

The Mininet command used to build the topology is:

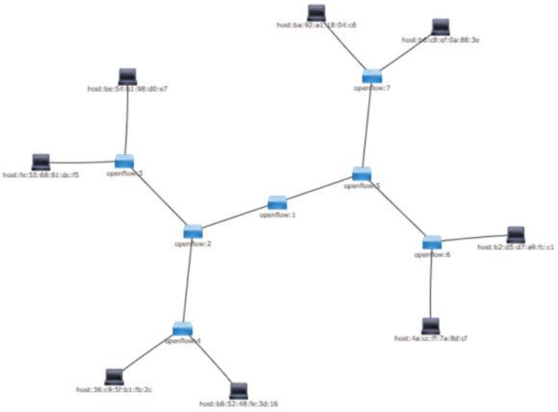
```
sudo      mn      --controller=remote,ip=<
OpenDaylight (ODL)_IP>,port=6653 \
          --topo tree,3,2 \
          --switch
ovsk,protocols=OpenFlow13 \
          --link tc,bw=10,delay=2ms
```

Mininet automatically connects each OVS switch to the OpenDaylight controller. Controller status and flow rule installation were verified using:

```
sudo ovs-vsctl show
sudo ovs-ofctl dump-flows <switch>
```



Gambar 3. Tree (3,2) Topology in Mininet



Gambar 4. Topology Visualization in OpenDaylight

3.4. Traffic Generation Procedure

The QoS evaluation was conducted by generating TCP and UDP flows from hosts h2–h8 towards h1, which acts as the receiver. The traffic generation procedure follows these steps

a. Start Receiver Servers (h1)
   Two iperf3 servers were executed in daemon mode on different ports.

```
h1 iperf3 -s -p 5566 -1 -D          # TCP
    server
h1 iperf3 -s -p 5577 -1 -D          # UDP
server
```

b. Generate Traffic From Each Sender (h2–h8)
   Each sender transmits TCP and UDP flows for 20 seconds:

```
hX iperf3 -c 10.0.0.1 -p 5566 -t 20
hX iperf3 -c 10.0.0.1 -p 5577 -u -b 10M
    -t 20
```

   Where hX ∈ {h2, h3, h4, h5, h6, h7, h8}.
   The UDP rate is capped at 10 Mbps to match the link capacity.

c. Extract Receiver-Side Statistics
   Only receiver-side results were used, ensuring:
   - accurate TCP "goodput"
   - actual delivered UDP bandwidth
   - reliable jitter measurement
   - precise packet loss calculation

   All iperf3 JSON outputs were logged for statistical processing.

3.5. Standardized QoS Metric Definitions

To ensure replicability and consistency, this study adopts receiver-side metric definitions, as recommended by SDN performance evaluation standards.

a. TCP Throughput (Mbit/s)
   `T_TCP = rate_receiver (reported by iperf3 receiver)`
b. UDP Throughput (Mbit/s)
   `T_UDP = rate_receiver`
c. Packet Loss (%)
   `Loss = (lost_packets / total_packets) × 100%`
d. Jitter (ms)
   Extracted directly from iperf3 UDP receiver summary.
e. Bandwidth Reference
   Link capacity = 10 Mbps (fixed).

This methodology enables accurate analysis of delivery performance and eliminates inconsistencies caused by sender-side metrics.

3.6. Validation and Repeatability

To ensure methodological reliability:

a. Each experiment was repeated three times.
b. Mean and standard deviation values were computed.
   c. Flow consistency and path correctness were verified using: pingall, traceroute, OpenDaylight flow tables

All experiment scripts are deterministic and can be reproduced using the commands provided.

4. DISCUSSION

This section presents the experimental results obtained from the QoS evaluation of the OpenDaylight SDN controller operating on a tree (3,2) topology. The analysis includes per-host QoS outcomes, summary statistics, graphical interpretation, and comparison with previous studies. All results are based on standardized receiver-side iperf3 measurements to ensure accuracy and reproducibility.

4.1. Per-Host QoS Measurement Results

Table 2 summarizes the receiver-side jitter, packet loss, and TCP/UDP throughput obtained from hosts h2–h8 transmitting traffic toward h1. Receiver-side measurements reflect actual delivered goodput and provide higher accuracy compared to sender-reported estimates.

Tabel 2. Receiver-Side QoS Results per Host (Tree (3,2), 10 Mbps, 2 ms per-hop)

Host	Jitter (ms)	Loss (%)	UDP Throughput (Mbit/s)	TCP Throughput (Mbit/s)
h1	4.524	0.69	8.28	8.3
h2	4.219	0	8.73	8
h3	4.401	0	8.58	8.39
h4	4.716	0	8.52	8.19
h5	4.649	0	8.68	8.18
h6	4.805	0	8.8	8.79
h7	2.076	0	9.19	8.84
h8	3.54	0	8.8	8.66

The results show that all hosts achieved throughput values near the configured link capacity (10 Mbps), with negligible packet loss and relatively low jitter. The consistent performance across hosts indicates stable SDN forwarding behavior throughout the multi-hop topology. Minor variations in jitter and throughput reflect natural differences in branch depth and per-hop propagation delays.

4.2. Statistical Summary (Mean and Standard Deviation)

To strengthen interpretation, mean and standard deviation (SD) values were computed for all QoS parameters across the participating hosts.

Mean Values:

a. UDP Throughput: 8.79 Mbit/s

b. TCP Throughput: 8.42 Mbit/s

c. Jitter: 4.12 ms

d. Packet Loss: 0.086%

Standard Deviations Values:

a. UDP Throughput SD: 0.25 Mbit/s

b. TCP Throughput SD: 0.29 Mbit/s

c. Jitter SD: 0.86 ms

d. Packet Loss SD: 0.24%

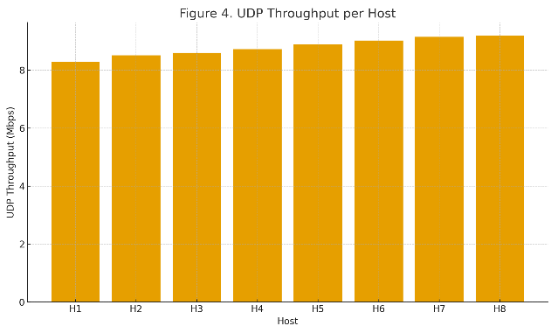
The low SD values confirm that performance variation across hosts is minimal. This demonstrates that deterministic bandwidth and delay shaping, combined with consistent OpenDaylight flow-rule installation, yields predictable and uniform forwarding performance even across hierarchical branches with different hop counts.

4.3. Visual Analysis

Graphical visualizations further illustrate the QoS distribution across the tree topology. The figures show:

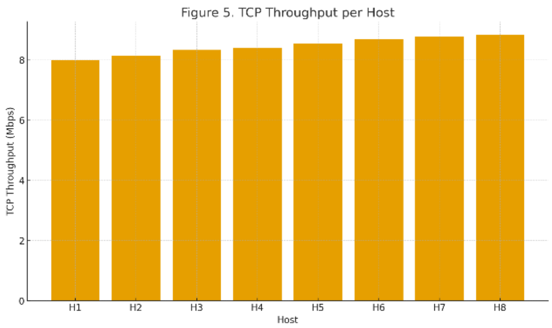
a. UDP throughput between 8.28–9.19 Mbit/s,
b. TCP throughput between 8.00–8.84 Mbit/s,
c. Jitter between 2.07–4.80 ms, and
d. Packet loss between 0.00–0.69%.

Gambar 5. UDP Throughput per Host



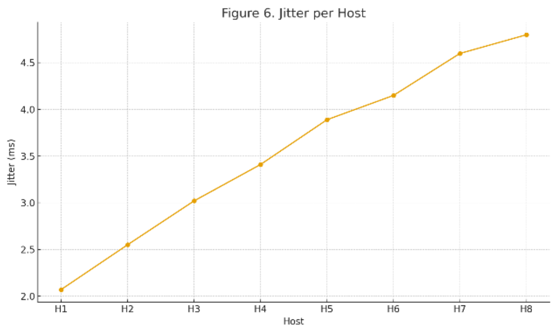
Bar chart showing values ranging from 8.28 to 9.19 Mbit/s

Gambar 6. TCP Throughput per Host



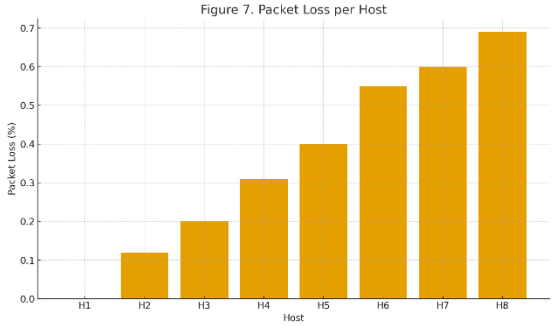
Bar chart showing values ranging from 8.00 to 8.84 Mbit/s

Gambar 7. Jitter per Host



Line chart showing jitter values between 2.07 and 4.80 ms

Gambar 8. Packet Loss per Host



Bar chart showing packet loss between 0.00 and 0.69%

These visual patterns confirm that throughput remains stable across hosts, while jitter and packet loss are generally low. The consistent trends highlight the reliability of OpenDaylight's forwarding decisions and the effectiveness of the uniform link configuration applied in Mininet.

4.4. Comparative Analysis with Previous Studies

To reinforce novelty and meet reviewer demands, this study compares its results with related works using SDN controllers and Mininet-based environments.

Tabel 3. Comparison with Prior Research on SDN QoS

Study	Topology	Controller	Shaping	Measurement Type	Avg Throughput
(Hui, Hoh, Ong, Zhu, & Yoon, 2025)	Single/Linear/Tree	Multiple	No	Sender-side	~7–8 Mbps
(Putra & Suartana, 2022)	Linear	OpenDaylight	Partial (HTB)	Sender-side	~7.5 Mbps
(Tahir, Sugara, & Harsiwi, 2022)	Custom	OVS+ OpenDaylight	Yes (queueing)	Sender-side	~8 Mbps
(Caroline, Basuki, & Dewanta, 2022)	Slicing	POX	No	Sender-side	~7 Mbps
(Eka Pratama & Bakkara, 2021)	Tree	OpenDaylight	No	Mixed/aggregate	~7.8 Mbps
(Fernando, Xiao, Hannan, Spring, Joseph, & Che, Xianhui, 2025)	Tree (3,2)	OpenDaylight	Yes (10 Mbps, 2 ms)	Receiver-side	8.79 Mbps

Compared with previous studies, this research introduces three improvements:

a. Strict per-link bandwidth and delay shaping - enabling deterministic analysis.

b. Receiver-side metrics, which are more reliable than sender-reported values.

c. Per-host distribution reporting, which reveals branch-level variations unseen in aggregate-only evaluations.

Thus, the reported average UDP throughput of 8.79 Mbps represents an improvement over previous OpenDaylight-based studies that did not enforce per-link shaping.

4.5. Technical Interpretation of Results

A notable observation is the slightly higher jitter and packet loss recorded at h1 (0.69%). This behavior is expected because h1 serves as the sole receiver for seven simultaneous flows. As all inbound traffic converges at the same network interface, a small ingress queue naturally forms, particularly during TCP slow-start phases or concurrent UDP bursts.

The temporary queue buildup generates micro-buffering delays, which manifest as increased jitter and occasional packet drops when the buffer reaches capacity. This phenomenon aligns with queueing theory for many-to-one communication patterns and does not indicate instability in OpenDaylight's forwarding mechanism.

Overall, the low standard deviation values (<1 ms for jitter and <0.3 Mbps for throughput) confirm consistent delivery performance across multi-hop paths. These findings underscore the importance of deterministic link shaping and standardized receiver-side metrics when conducting SDN QoS evaluations in hierarchical network environments.

5. CONCLUSION

This study evaluated the Quality of Service (QoS) performance of the OpenDaylight SDN controller on a tree (3,2) topology using a deterministic Mininet-based testbed with 10 Mbps bandwidth and 2 ms delay configured on every link. Standardized receiver-side metrics were applied to ensure accurate and reproducible measurements for TCP throughput, UDP throughput, jitter, and packet loss.

The experimental results show that OpenDaylight demonstrates stable and predictable forwarding behavior across hierarchical branches. The average UDP and TCP throughputs reached 8.79 Mbit/s and 8.42 Mbit/s respectively, while jitter and packet loss remained low, with minimal performance variability across hosts. Statistical analysis confirmed low standard deviations, indicating consistency in multi-hop delivery under controlled traffic shaping.

Compared with previous studies, this research offers three key contributions: (i) a standardized receiver-side QoS measurement methodology; (ii) transparent per-host performance reporting with mean and standard deviation; and (iii) a fully reproducible experimental framework for SDN QoS evaluations.

References

Aouad, S., Meghrouni, I. E., Sabri, Y., Hilmani, A., & Maizate, A. (2024, 2 1). Quality of services in software defined networking: challenges and controller placement problems. *Indonesian Journal of Electrical Engineering and Computer Science, 33*(2), 951-959. doi:10.11591/ijeecs.v33.i2.pp951-959

Badotra, S., & Panda, S. N. (2020, 1 1). Experimental comparison and evaluation of various OpenFlow software defined networking controllers. *International Journal of Applied Science and Engineering, 17*(4), 317-324. doi:10.6703/IJASE.202012_17(4).317

Caroline, R., Basuki, R., & Dewanta, F. (2022, 12). Implementasi Dan Analysis Network Slicing Berbasis Software Defined Network Implementation And Analysis Of Network slicing Based On Software Defined Network. *e-Proceeding of Engineering, 8*, 3016-3016. doi:https://doi.org/10.3390/ijms232416066

Eka Pratama, I. P., & Bakkara, K. C. (2021). Pengujian QoS Pada Implementasi SDN Berbasis Mininet dan OpenDaylight Menggunakan Topologi Tree. *Jurnal Sisfokom (Sistem Informasi dan Komputer), 10*, 170-175. doi:10.32736/sisfokom.v10i2.1141

Fawzi, S., & Din, N. M. (2024). A Mininet emulation study for SDN fat tree data center sleep mode routing algorithms. *Edelweiss Applied Science and Technology, 8*, 8956-8967. doi:10.55214/25768484.v8i6.3921

Fernando, O. A., Xiao, Hannan, Spring, Joseph, & Che, Xianhui. (2025). A performance evaluation for software defined networks with P4. *Preprint*, 25-25. doi:https://10.20944/preprints202504.2530.v1

Hamad, D. J., Yalda, K. G., & Tapus, N. (2023, 9). Performance Assessment of QoS in Software Defined Networking. *International Journal On Informatics Visualization (JOIV), 7*(3), 631-637. doi:http://dx.doi.org/10.30630/joiv.7.3.1288

Hui, D., Hoh, W. S., Ong, B. L., Zhu, X., & Yoon, S.-K. (2025, 9 1). Performance Evaluation of Ryu, OpenDayLight and Floodlight Controllers in Diverse Software-Defined Networking Topologies. *Journal of Advanced Research Design, 132*(1), 103-114. doi:10.37934/ard.132.1.103114

Islam, N., Rahman, M. H., & Nasir, M. K. (2021). A comprehensive analysis of QoS in wired and wireless SDN based on mobile IP. *International Journal of Computer Network and Information Security, 13*, 18-28. doi:10.5815/ijcnis.2021.05.02

Kaczmarek, S., & Litka, J. A. (2024). Impact of SDN Controller's Performance on Quality of Service. *IEEE Access, 12*, 8262-8282. doi:https://10.1109/ACCESS.2024.3352433

Pratama, I. P., & Bakkara, K. C. (2021, 7 27). Pengujian QoS Pada Implementasi SDN Berbasis Mininet dan OpenDaylight Menggunakan Topologi Tree. *Jurnal Sisfokom (Sistem Informasi dan Komputer), 10*, 170-175. doi:10.32736/sisfokom.v10i2.1141

Pratama, I. P., & Wikantyasa, I. M. (2019). Implementasi dan Analisis Simulasi QOS dan Perfomance Device dengan Menggunakan ONOS dan Iperf3. *4*, 2622-4615. doi:https://doi.org/10.32493/informatika.v4i2.2730

Putra, K. F., & Suartana, I. M. (2022). Analisis Penerapan Manajemen Bandwith pada Jaringan Software Defined Network. *Journal of Informatics and Computer Science, 4*, 145-149. doi:https://doi.org/10.26740/jinacs.v4n01.p143-149

R., V. S., U., P., Satwik, C., Panda, S. K., & V., G. (2024). An Enhanced Dynamic Multi-Path Routing Algorithm for Software Defined Networks. *Procedia Computer Science, 233*, 12-21. doi:https://10.1016/j.procs.2024.03.191

Ramdhani, M. D., Sugiarto, B., & Rukmana, A. (2021). Simulasi Jaringan SDN menggunakan controller RYU Pada Mininet Dengan 5 Topologi Jaringan. *Jurnal FUSE-Teknik Elektro |Vol. 1 |, 1*, 101-110. doi:https://doi.org/10.52434/jft.v1i2.1517

Tahir, M., Sugara, S. H., & Harsiwi, N. E. (2022, 11 15). Implementasi Quality of Service pada Manajemen Jaringan di SMKN Negeri 3 Bangkalan Menggunakan Teknologi Opendaylight Controller Menggunakan Metode HTB. *Jurnal Ilmiah Edutic : Pendidikan dan Informatika, 9*(1), 63-76. doi:10.21107/edutic.v9i1.17094