Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16086
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Comparative Study of Johnson and Bellman-Ford for Shortest Path in OpenFlow SDN

Afriza Tri Rizki[1)*], Funny Farady Coastera[2)], Ernawati[3)]
[1,2,3)] Informatics Study Program, Faculty of Engineering, University of Bengkulu, Bengkulu, Indonesia
[1)]afrizatririzki@email.com, ffaradyc@unib.ac.id, ernawati@unib.ac.id

Submitted :12 April 2026 | **Accepted** : May 9, 2025 | **Published** : July 6, 2026

Abstract: Software-Defined Networking (SDN) provides a highly programmable architecture specifically by decoupling the control plane from the data plane. The efficiency of the controller in mapping topologies and responding to link failures depends heavily on the shortest-path algorithm used. While previous studies have evaluated algorithms like Bellman-Ford in small scale SDN, empirical comparisons with hybrid algorithms like Johnson in medium-scale dense topologies remain significantly limited. This study aims to provide an empirical comparative evaluation of Johnson and Bellman-Ford algorithms on OpenFlow 1.3 using the RYU controller, analyzing scalability across ring (sparse) and full-mesh (dense) topologies from 10 to 50 nodes. The research methodology relies on experimental emulation using Mininet to test convergence time, throughput, and recovery time during dynamic link failures. The results indicate that in sparse ring topologies, both algorithms achieve similar convergence under 0,06 seconds. However, in dense 50 node full-mesh networks containing 2.450 links, Bellman-Ford demonstrates a faster average convergence of 37,93 seconds compared to Johnson's 47,44 seconds, primarily due to the absence of graph reweighting overhead, despite exhibiting higher variance. Both algorithms maintained stable throughput, and while recovery times generally met the near carrier-grade standard, some scenarios in dense networks reached 60 milliseconds, slightly exceeding the 50 ms threshold. This study evaluates recovery during single link failure scenarios. In conclusion, Bellman-Ford is highly recommended for dense data center infrastructures, while Johnson is optimal for sparse networks requiring instant route recovery.

Keywords: Bellman-Ford Algorithm, Johnson Algorithm, OpenFlow, Shortest Path, Software-Defined Networking

INTRODUCTION

Software-Defined Networking (SDN) has emerged as a promising paradigm for simplifying network management by logically decoupling the control plane from the data plane. This separation provides centralized, proactive routing flexibility through protocols such as OpenFlow, which has been proven to significantly boost network resilience against traffic congestion (Islam et al., 2024). In OpenFlow based SDN, the controller not only maintains a global view of the network but also determines how flows are forwarded by installing rules into switches. The efficiency of the controller in mapping the topology and responding to link failures depends entirely on the embedded shortest-path algorithm (Patel et al., 2024).

While previous studies have widely explored and evaluated the performance parameters of conventional algorithms like Bellman-Ford in small scale SDN environments (Abed & Noaman, 2025; Ghimire & Basnet, 2023). The current literature still lacks a comprehensive comparative evaluation against hybrid algorithms like Johnson, particularly when subjected to the extremes of medium scale-dense graphs up to 50 nodes (Mahdi & Mahmood, 2025). This research gap is critical. Theoretically, Johnson's algorithm possesses a computational advantage in sparse graphs by combining Bellman-Ford and Dijkstra to compute all-pairs shortest paths efficiently. However, its resilience against control plane overhead specifically convergence anomalies and recovery latency in highly dense full-mesh topologies compared to the pure iterative execution of Bellman-Ford has not been empirically proven.

To address this issue, this study contributes by providing an in-depth empirical comparative evaluation of Johnson and Bellman-Ford algorithms operating on the OpenFlow 1.3 standard using the RYU controller. Furthermore, this research extensively analyzes the scalability limits across distinct architectural extremes, namely the sparse ring topology and the highly dense full-mesh topology scaled at 10, 30, and 50 nodes. Finally, the

e-ISSN : 2541-2019
p-ISSN : 2541-044X

findings from these evaluations are synthesized to formulate evidence-based practical guidelines for data center architecture, ensuring that network administrators and engineers can select the most optimal routing mechanism to maintain continuous and reliable data transmission under various structural complexities.

LITERATURE REVIEW

Studies regarding the optimization of routing algorithms in SDN continue to evolve rapidly. In link failure scenarios, Chiesa et al. (2021) highlight that fast-recovery mechanisms operating below the 50 millisecond threshold (carrier-grade) are an absolute industry standard. This is further supported by Hainana et al. (2023), who implemented a dedicated NFV Middlebox to accelerate failure detection in core networks, although this approach introduces external architectural dependencies. Achieving minimal latency is highly influenced by network topology complexity. Furthermore, routing environments often face fuzzy or uncertain conditions, where Bellman-Ford has been mathematically proven by Parimala et al. (2021) to provide a robust foundation for complex route computations. To handle network resilience proactively, modern approaches such as Graph Neural Networks have been introduced by Islam et al. (2024), yet they often incur high computational overhead compared to deterministic algorithms.

Recent studies conducting comparative evaluations between Bellman-Ford and Dijkstra in SDN affirm that algorithm execution directly impacts controller overhead (Tammanashastri et al., 2024). Furthermore, evaluations by Ghimire and Basnet (2023) on Mininet demonstrate the varying performance of conventional shortest-path algorithms. However, previous studies and comprehensive reviews on network routing algorithms (Mahdi & Mahmood, 2025) have not isolated the link density variable between ring and full-mesh topologies to test the upper bound limits of the hybrid Johnson algorithm. Table 1 maps the existing research gaps and the positioning of this study.

Table 1. State of the Art and Research Gap Mapping

Author & Year	**Study Focus**	**Limitations/Gaps**	**This Research Position**
Parimala et al. (2021)	Bellman-Ford optimization under picture fuzzy environment.	Focuses on mathematical modeling without practical SDN controller implementation.	Validates Bellman-Ford's robustness practically within an OpenFlow 1.3 SDN architecture.
Chiesa et al. (2021)	Fast-recovery mechanisms in packet-switched networks.	Does not specifically compare routing algorithms in the RYU controller environment.	Adopts the <50ms standard as the baseline for recovery time evaluation.
Hainana et al. (2023)	NFV Middlebox for link failure detection and recovery in SDN.	Relies on an additional Middlebox, adding architectural complexity.	Uses native RYU controller capabilities for link failure recovery without external dependencies.
Ghimire & Basnet (2023)	Shortest path routing performance evaluation in Mininet.	Focuses solely on simple, static topologies using conventional algorithms.	Expands topology evaluation up to a 50-node Full-Mesh scale comparing hybrid algorithms.
Islam et al. (2024)	Resilient proactive routing using Deep Q-Networks.	High computational overhead for AI-based routing models.	Proposes a lightweight, deterministic algorithmic approach for rapid failover.

Author & Year	Study Focus	Limitations/Gaps	This Research Position
Tammanashastri et al. (2024)	Comparative analysis of Bellman-Ford and Dijkstra in SDN.	Does not test the efficiency of hybrid algorithms (Johnson) under extreme link density.	Conducts a head-to-head empirical comparison specifically between Bellman-Ford and Johnson.

METHOD

This study utilizes a quantitative experimental emulation approach. Mininet and the RYU controller were selected because this combination accurately represents the industry-standard OpenFlow 1.3 architecture (Lantz & O'Connor, 2015; Patel et al., 2024). To ensure data robustness, experiments were not limited to a single configuration. Instead, cross-testing was conducted on two graph extremes: the Ring topology, representing a sparse graph with minimum structural complexity, and the Full-Mesh topology, representing a dense graph with maximum exponential links. Tests were incrementally scaled at 10, 30, and 50 nodes. Link failure scenarios were dynamically injected during active data transmission to measure failover latency under realistic network stress.

RESULT

The experimental data were directly extracted from the RYU controller and Mininet system logs. Three main metrics were measured in these experiments, namely convergence time, recovery time, and throughput.

Ring Topology Results

Table 2 presents the detailed experimental results for iterations 1 to 3, along with their average values, for the ring topology using the Johnson and Bellman-Ford algorithms.

Table 2. Experimental Results - Ring Topology (Sparse Graph)

Scale (Nodes)	Algorithm	Metric	Iteration 1	Iteration 2	Iteration 3	Average	Std. Dev.
10 Nodes	**Johnson**	Convergence (s)	0,0336	0,0377	0,0328	**0,0347**	0,0026
		Throughput (Mbps)	23.212	21.134	22.505	**22.283**	1,057
		Recovery (s)	0,0126	0,0125	0,0138	**0,0129**	-
	Bellman-Ford	Convergence (s)	0,0305	0,0410	0,0245	**0,0320**	0,0084
		Throughput (Mbps)	22.701	23.558	22.722	**22.993**	0,489
		Recovery (s)	0,0213	0,0202	0,0198	**0,0204**	-
30 Nodes	**Johnson**	Convergence (s)	0,0345	0,0356	0,0487	**0,0396**	0,0079
		Throughput (Mbps)	21.609	23.223	23.532	**22.788**	1,033
		Recovery (s)	0,0142	0,0152	0,0161	**0,0151**	-
	Bellman-Ford	Convergence (s)	0,0504	0,0481	0,0397	**0,0460**	0,0056
		Throughput (Mbps)	23.284	23.656	23.452	**23.464**	0,186
		Recovery (s)	0,0171	0,0175	0,0122	**0,0156**	-
50 Nodes	**Johnson**	Convergence (s)	0,0522	0,0487	0,0532	**0,0513**	0,0024
		Throughput (Mbps)	21.926	22.611	23.422	**22.653**	0,749
		Recovery (s)	0,0202	0,0250	0,0192	**0,0214**	-
	Bellman-Ford	Convergence (s)	0,0605	0,0361	0,0527	**0,0497**	0,0125
		Throughput (Mbps)	22.063	22.196	21.199	**21.819**	0,541
		Recovery (s)	0,0218	0,0192	0,0225	**0,0212**	-

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16086
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Based on Table 2, for the ring topology at a scale of 10 nodes, Bellman-Ford achieves a slightly faster average convergence time of 0,0320 seconds compared to Johnson at 0,0347 seconds. However, Johnson demonstrates more consistent performance with a lower standard deviation of 0,0026 seconds compared to Bellman-Ford's 0,0084 seconds. At 30 nodes scale, Johnson becomes marginally faster with convergence time of 0,0396 seconds compared to Bellman-Ford at 0,0460 seconds, with both algorithms showing comparable stability (SD of 0,0079 and 0,0056 seconds respectively). Interestingly, at 50 nodes scale, Bellman-Ford again achieves slightly faster convergence (0,0497 seconds) compared to Johnson (0,0513 seconds), while Johnson maintains significantly better stability with SD of 0,0024 seconds versus Bellman-Ford's 0,0125 seconds. Throughput for both algorithms remains relatively stable in the range of approximately 21.000 to 23.000 Mbps with low standard deviation (<1.1 Mbps) across all scales, with Bellman-Ford demonstrating slightly better throughput consistency (SD ranging from 0.186 to 0,541 Mbps) compared to Johnson (SD ranging from 0,749 to 1,057 Mbps). This indicates that low structural complexity does not impose a significant burden on data forwarding performance (Patel et al., 2024).

Full Mesh Topology Results

Table 3 presents the detailed experimental results for the full-mesh network topology. The dense link characteristics of this topology pose computational challenges, which are reflected in higher convergence durations.

Table 3. Experimental Results - Mesh Topology (Dense Graph)

Scale (Nodes)	Algorithm	Metric	Iteration 1	Iteration 2	Iteration 3	Average	Std. Dev.
10 Nodes	**Johnson**	Convergence (s)	0,0505	0,0526	0,0465	**0,0498**	0,00
		Throughput (Mbps)	24.586	21.740	22.781	**23.035**	1,4
		Recovery (s)	0,0400	0,0300	0,0400	**0,0366**	-
	Bellman-Ford	Convergence (s)	0,0507	0,0373	0,0512	**0,0464**	0,01
		Throughput (Mbps)	23.362	22.202	22.067	**22.543**	0,7
		Recovery (s)	0,0400	0,0300	0,0400	**0,0366**	-
30 Nodes	**Johnson**	Convergence (s)	0,1729	0,1087	0,0884	**0,1233**	0,04
		Throughput (Mbps)	22.795	21.838	23.580	**22.737**	0,9
		Recovery (s)	0,0500	0,0300	0,0300	**0,0366**	-
	Bellman-Ford	Convergence (s)	0,0908	0,0713	0,0865	**0,0828**	0,01
		Throughput (Mbps)	24.054	23.925	22.001	**23.326**	1,2
		Recovery (s)	0,0500	0,0300	0,0400	**0,0400**	-
50 Nodes	**Johnson**	Convergence (s)	31,9149	60,4023	50,0267	**47,4479**	14,42
		Throughput (Mbps)	22.015	22.496	23.682	**22.731**	8.581,0
		Recovery (s)	0,0500	0,0400	0,0600	**0,0500**	-
	Bellman-Ford	Convergence (s)	14,4462	18,0657	81,2959	**37,9359**	37,60
		Throughput (Mbps)	24.618	19.606	26.243	**23.489**	34.581,0
		Recovery (s)	0,0500	0,0400	0,0500	**0,0466**	-

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16086
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 3 shows that for the mesh topology at a scale of 10 nodes, both algorithms still achieve convergence in under one second with excellent stability, where Johnson demonstrates a standard deviation of 0,0031 seconds and Bellman-Ford 0.0079 seconds. At the 30 node scale, convergence performance remains acceptable albeit slower than the ring topology, with Bellman-Ford showing better stability (SD 0,0103 seconds) compared to Johnson (SD 0,0441 seconds). However, at the 50 node scale, the extreme complexity of the full mesh topology containing 2.450 links leads to a dramatic increase in convergence time, shifting significantly away from rapid performance. Both algorithms experienced exponential delays, where Johnson reached an average of 47,44 seconds and Bellman-Ford 37,93 seconds. Notably, Johnson maintains significantly better stability with a standard deviation of 14,42 seconds compared to Bellman-Ford's substantially higher variance of 37,60 seconds, indicating that Bellman-Ford's performance becomes highly unpredictable in dense network environments. Similarly, throughput variance escalates dramatically at 50 nodes, with Johnson's standard deviation reaching 8.581 Mbps and Bellman-Ford's soaring to 34.581 Mbps, further confirming the challenges of maintaining consistent performance in highly connected topologies (Mahdi & Mahmood, 2025). To provide a clearer comparative visualization, the convergence trends for both topologies are depicted in Fig. 1. The line graph illustrates the stable, rapid convergence of the ring topology contrasted sharply with the exponential delay spike in the 50 node mesh topology.

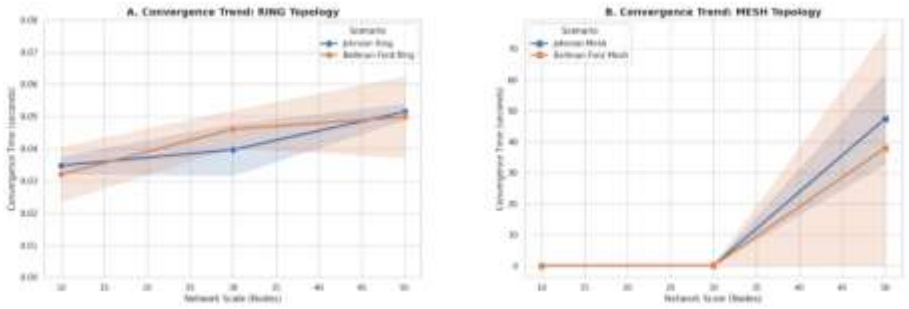

Fig. 1 Convergence Time Trends for Ring Topology (Panel A) and Mesh Topology (Panel B)

Regarding data plane performance, the throughput stability is presented in the heatmap in Fig. 2. The visualization confirms that the average throughput is highly clustered in the higher bandwidth spectrum across both algorithms, highlighting consistent forwarding capacity regardless of the routing module used.

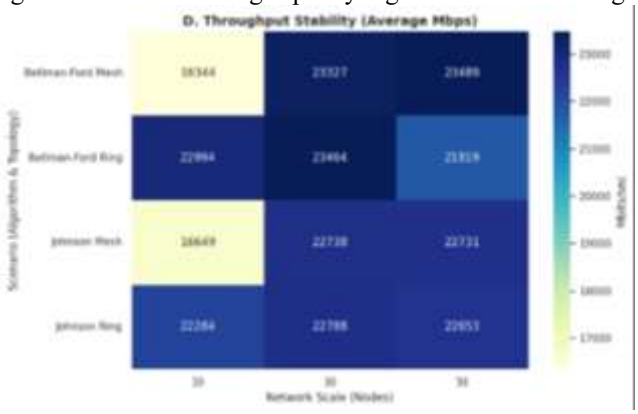

Fig. 2 Average Throughput Stability Across Various Scenarios

Additionally, the system's responsiveness to network disruptions is illustrated in Fig. 3. The scatter plot with confidence intervals demonstrates the recovery efficiency during link failures. The data points show that recovery in the ring topology remains tightly clustered with low variance, whereas the mesh topology exhibits higher variance and latency, though still operating within acceptable thresholds.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16086
e-ISSN : 2541-2019
p-ISSN : 2541-044X

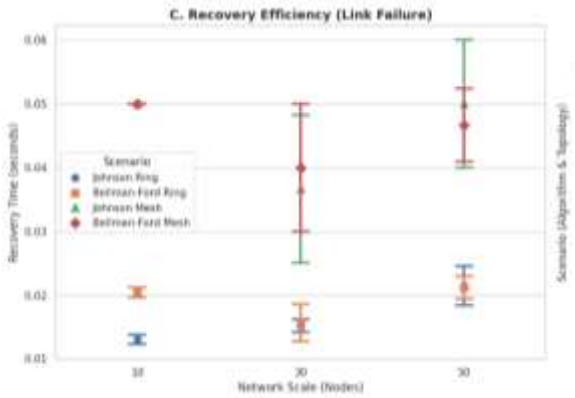

Fig. 3 Recovery Efficiency (Link Failure) with Confidence Intervals

To validate the statistical significance of the convergence time differences between Johnson and Bellman-Ford, an independent samples t-test was conducted across all scenarios. For the ring topology, the calculated p-values were 0,62 (10 nodes), 0,31 (30 nodes), and 0,83 (50 nodes). In the 50 node mesh topology, the p-value was 0,72. Since all p-values are greater than the 0,05 significancee level, it is confirmed that there is no statistically significant difference in convergence performance between the two algorithms. The differences in average values are primarily attributed to high intra-iteration variance rather than absolute algorithmic superiority.

DISCUSSIONS

The experimental results reveal performance anomalies that align with graph algorithm complexity theories. In the sparse ring topology, the performance of Johnson and Bellman-Ford is virtually identical, achieving rapid convergence below 0,06 seconds. However, as visualized in Fig. 1, a breaking point occurs in the 50 node full-mesh topology. Johnson's convergence time spiked dramatically to 47,44 seconds, lagging significantly behind Bellman-Ford at 37,93 seconds. Theoretically, this is justifiable as Johnson's algorithm must perform pre-processing graph reweighting to eliminate negative edges before executing Dijkstra for every node (Mahdi & Mahmood, 2025). In a dense mesh, this pre-processing overhead creates a computational bottleneck on the RYU controller's CPU cycles. Conversely, while its single-source iterative nature avoids the reweighting phase (Parimala et al., 2021; Saxena et al., 2023), Bellman-Ford's performance is highly topology-dependent. It generally provides faster average computation in dense networks; however, it exhibits significant instability in these environments, as evidenced by the high standard deviation (SD = 37,60) in the 50 node mesh scenario. In contrast, within the sparse ring topology, Bellman-Ford remains highly stable with a very low variance (SD = 0,0125). Regarding data plane performance, the throughput stability is quantitatively confirmed by the low standard deviation across all scales (SD < 1.5 Mbps in the ring topology), proving that control plane overhead does not disrupt bulk forwarding. As depicted in the heatmap in Fig. 2, post-convergence throughput metrics remained robust between 21.000 and 23.500 Mbps (Hardin et al., 2023). This confirms previous SDN performance studies (Linsheng et al., 2022) and provides a critical analysis that routing algorithm efficiency is purely a control plane optimization issue.

Once flow rules are installed in the Open vSwitch, bulk data forwarding is autonomously handled by the data plane (Islam et al., 2024). In failover testing, as explicitly shown in Fig. 3, Johnson recorded slightly superior recovery times in the ring topology (Yasin, 2022), proving that in sparse environments, instant substitution paths are computed more efficiently by hybrid algorithms. Regarding recovery time in the dense mesh, while most results fall within the carrier-grade threshold (<50 ms) (Chiesa et al., 2021; Nurwarsito & Prasetyo, 2023), the occurrence of 60 ms latency in Johnson's 50-node iteration suggests that additional optimization is required to maintain this standard consistently in highly dense architectures. This failover speed aligns with link failure detection evaluations in core SDN architectures (Hainana et al., 2023). Overall, this study focuses on medium-scale topologies up to 50 nodes, providing a comparative baseline rather than a new algorithmic novelty.

Although the experiments yielded consistent results, this study acknowledges certain limitations. The evaluation was conducted in an emulated environment using Mininet, which may not fully capture the physical hardware bottlenecks of enterprise grade servers. Furthermore, the testing scope was limited to static node configurations and single link failure scenarios. Future research should address these threats to validity by evaluating synchronization latency in multi controller SDN environments using dynamic, multi link failure simulations.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

CONCLUSION

This study provides a comprehensive empirical evaluation regarding the scalability trade-offs between Johnson and Bellman-Ford algorithms in OpenFlow SDN environments. The findings confirm that topology density is the primary determinant of algorithm performance. In sparse graph environments, Johnson's algorithm is recommended due to its highly responsive recovery latency. However, for medium-scale, dense data center infrastructures utilizing full-mesh topologies, Bellman-Ford is recommended as it provides a faster average convergence computation for the control plane, despite exhibiting higher variance. Practically, network engineers can use these findings as a baseline to select the appropriate routing module based on physical data center architecture to maintain optimal post-convergence transmission.

REFERENCES

Abed, S. S., & Noaman, S. F. (2025). *Comparative Analysis of Shortest Path Algorithms*. *4150*(5), 131–143.

Chiesa, M., Kamisi, A., Rak, J., & Member, S. (2021). *A Survey of Fast-Recovery Mechanisms in Packet-Switched Networks*. *23*(2), 1253–1301. https://doi.org/10.1109/COMST.2021.3063980

Ghimire, R., & Basnet, R. K. (2023.). *Shortest Path Routing Performance Evaluation over SDN Environment*. *5*(4), 405–422.

Hainana, H. (2023). Design of a NFV Traffic Engineering Middlebox for Efficient Link Failure Detection and Recovery in SDN Core Networks. *2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, 1–7. https://doi.org/10.1109/ETNCC59188.2023.10284948

Hardin, B., Comer, D., & Rastegarnia, A. (2023). *On the Unreliability of Network Simulation Results FROM Mininet and iPerf*. *12*(1), 1–9. https://doi.org/10.18178/ijfcc.2023.12.1.596

Islam, A., Atat, R., & Member, S. (2024). Software-Defined Networking-Based Resilient Proactive Routing in Smart Grids Using Graph Neural Networks and Deep Q-Networks. *IEEE Access*, *12*(June), 111169–111186. https://doi.org/10.1109/ACCESS.2024.3438938

Lantz, B., & Connor, B. O. (2015). *A Mininet-based Virtual Testbed for Distributed SDN Development*. 365–366.

Linsheng, R., Derahman, M. N., Kadir, M. F. A., Mohamed, M. A., & Kamarudin, S. (2022). *International Journal of Advanced and Applied Sciences The performance effect due to varying network topologies on a software-defined network employing the k-shortest path*. *9*(6), 134–144.

Mahdi, H., & Mahmood, I. (2025). *A Comprehensive Review of Shortest Path Algorithms for Network Routing*. *18*(3), 152–175.

Nurwarsito, H., & Prasetyo, G. (2023). *Implementation Failure Recovery Mechanism using VLAN ID in Software Defined Networks*. *14*(1), 709–714.

Parimala, M., Broumi, S., Prakash, K., & Topal, S. (2021). Bellman – Ford algorithm for solving shortest path problem of a network under picture fuzzy environment. *Complex & Intelligent Systems*, *7*(5), 2373–2381. https://doi.org/10.1007/s40747-021-00430-w

Patel, K. P., Chaudhari, J. P., Mewada, H. K., Jayswal, H. S., & Patel, R. V. (2024). *Shortest Path Forwarding in Software-Defined Networks Using RYU Controller*. *11*(5), 299–305.

Saxena, M. C., Sabharwal, M., & Bajaj, P. (2023). *An Optimised Shortest Path Algorithm for Network Rotuting & SDN Improvement on Bellman-Ford Algorithm*. *June*, 20–31.

Tammanashastri, P. R. (2024). Comparative Analysis of Bellman-Ford and Dijkstra Algorithms in Software Defined Networking. *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, 1–8. https://doi.org/10.1109/ICDICI62993.2024.10810921

Yasin, Q. (2022). *Reliable Multipath Flow for Link Failure Recovery in 5G Networks Using SDN Paradigm*. https://doi.org/10.5755/j01.itc.51.1.29408