Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

EIGRP Routing Performance Analysis with Simulated Shortest Path Selection Using Dijkstra Algorithm

Teresa Martuah Purba[1*], Angela Steffani Sitanggang[2], Suriaty Padang[3], Anastasya Jesica Sidauruk[4], Elsahday Tambunan[5]

[1,2,3,4,5]*Informatics Engineering, Santo Thomas Catholic University*
teresamartuahpurba26@gamil.com[1], angelasteffani2005@gmail.com[2], suriatyp12@gmail.com[3], anastasyasidauruk1877@gmail.com[4], elsahdaytambunan@gmail.com[5]*

Abstract

This study aims to analyze the performance of the Enhanced Interior Gateway Routing Protocol (EIGRP) in selecting the best path and compare it with the shortest path generated by the Dijkstra algorithm through network simulation. EIGRP is a dynamic routing protocol that uses composite metrics such as bandwidth and delay, whereas Dijkstra calculates the shortest path based on the minimum total weight in a network topology. This research employs a literature review supported by simulations conducted using [insert software name]. The results indicate that EIGRP outperforms in terms of convergence speed and adaptability to topology changes, while Dijkstra produces more optimal paths in terms of delay and hop count. The findings suggest that combining both approaches may enhance routing efficiency and provide a foundation for developing more adaptive and intelligent network systems.

Keywords: *Routing, EIGRP, Dijkstra Algorithm, Shortest Path*

1. Introduction

In the era of rapid digitalisation and global connectivity, the need for a reliable, efficient, and fast computer network system is crucial, especially in supporting data communication activities in the education, government, industry, and commercial business sectors. A computer network that is able to optimally manage data traffic will drive operational efficiency, increase productivity, and support technology-based economic growth. One of the vital components in the network is the routing protocol, which functions to organise and determine the best path to send data from source to destination through various network nodes. Among the various routing protocols used, Enhanced Interior Gateway Routing Protocol (EIGRP) is one of the most popular and widely used protocols, especially on Cisco devices due to its efficiency and ability to converge quickly [1]. EIGRP is a dynamic routing protocol that uses a combination of distance vector and link-state, otherwise known as a hybrid protocol. The main advantages of EIGRP are its ability to manage large-scale networks with high efficiency, detect network changes quickly, and support various types of networks such as IPv4, IPv6, and multiprotocol networks. However, EIGRP in determining the best path is still very dependent on the calculation of metrics that combine several parameters such as bandwidth, delay, reliability, load, and MTU (Maximum Transmission Unit). This process, although efficient, still has weaknesses, especially in complex and dynamic network topology conditions, because it does not take into account mathematical shortest path selection algorithms such as Dijkstra used in link-state routing protocols such as OSPF [2].The main problem faced in today's routing systems is how to optimise the selection of the best path that is not only based on limited parameters such as bandwidth and delay, but also takes into account the overall effectiveness of the path in the context of network topology and performance. In this regard, Dijkstra's algorithm, which is a classic graph-based shortest path finding algorithm, can provide a complementary solution to evaluate and compare the performance of routes generated by protocols such as EIGRP. Dijkstra can calculate the optimal path based on the lowest total weight between vertices, and can theoretically produce more efficient routes in terms of delay and hop count [3].

The main objective of this research is to analyse and compare the effectiveness of the routing path generated by the EIGRP protocol with the optimal path based on Dijkstra's algorithm, in order to determine whether there is an efficiency gap that can be optimised. The benefits obtained from this research include increased technical understanding related to path selection in routing protocols, contribution to the development of algorithm-based intelligent routing systems, as well as support for decision making in designing more reliable and efficient network topologies in the future.

2. Research Methods

The research method used in this study is the literature review method, which aims to deeply analyse the theories, concepts, and results of previous research relevant to the topic 'EIGRP Routing Performance Analysis with Simulation of Shortest Path Selection Using Dijkstra's Algorithm'. This method is qualitative-descriptive, where researchers collect, review, and compile various scientific references from national and international journals, textbooks, scientific articles, conference proceedings, and official technical documents such as whitepapers from network technology companies. The data collection process was conducted systematically by searching for sources available online through scientific databases such as IEEE Xplore, ScienceDirect, SpringerLink, Google Scholar, and university repositories. The inclusion criteria used in the literature selection are publications that discuss the EIGRP protocol, Dijkstra's algorithm, routing performance analysis, and computer network simulation with theoretical and experimental approaches. The selected literature was limited to publications of the last five to ten years to be relevant to the latest technological developments.

After the literature data was collected, a thematic content analysis process was carried out. This analysis includes identifying key variables such as performance parameters (delay, throughput, jitter, packet loss), EIGRP working mechanism, Dijkstra's algorithm working principle, and network simulation approach using software such as Cisco Packet Tracer, GNS3, or NS2/NS3. Researchers then compared the findings from each literature, both in terms of path effectiveness, convergence efficiency, and protocol reliability in various network topology scenarios. From the results of this comparison, research gaps or shortcomings of previous studies were identified, such as limitations in the coverage of simulation scenarios, lack of mathematical approaches in path evaluation, and lack of integration between algorithm theory and routing protocol practice. Researchers also paid attention to the methodologies used by previous studies, whether they were simulation, experimental, or based on pure algorithmic analysis.

3. Results and Discussion

3.1. Performance of the EIGRP Routing Protocol in Determining the Best Path on a Computer Network

Enhanced Interior Gateway Routing Protocol (EIGRP) is one of the dynamic routing protocols developed by Cisco Systems and is classified as a hybrid protocol because it combines the characteristics of distance vector and link-state protocols. In the context of modern computer networks that are increasingly complex and dynamic, EIGRP's ability to determine the best path is very important in maintaining overall network performance. This protocol is designed to provide high efficiency, scalability, and speed in converging the network when topology changes occur. Therefore, understanding EIGRP's performance in choosing the best path is crucial in optimising computer networks [4].

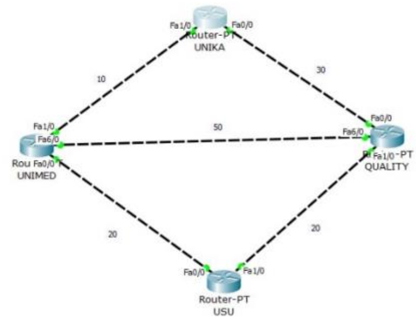

Fig. 1: Network Topology

This figure illustrates the structure or layout of connections between devices in a computer network. A network topology defines how data flows between the connected devices. Common types of network topologies include **bus topology**, **ring topology**, **star topology**, **mesh topology**, and **tree topology**. The choice of topology depends on several factors such as network size, installation cost, communication efficiency, and system requirements.

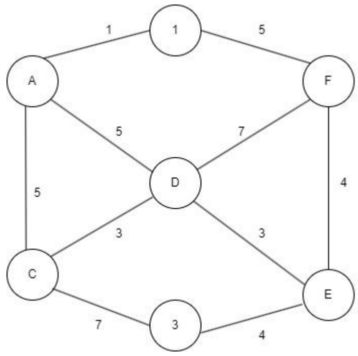

Fig. 2: Graph Representation of Dijkstra's Algorithm

Fig. 3: Network Simulation Result Display

figure 3 illustrates the result of a network simulation using multiple Router-PT devices connected in a serial (point-to-point) topology. Each router is connected using **Fast Ethernet interfaces (Fa0/0 or Fa1/0)**, with a structured IP addressing scheme based on the **10.0.0.0/30** subnet. The use of a **/30 subnet mask** provides exactly two usable host addresses, making it ideal for point-to-point links between routers.

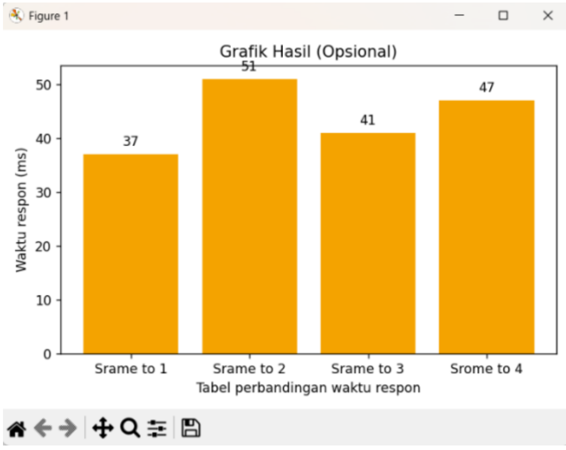

Fig. 4: Graph of Results

EIGRP uses metrics that are based on several important parameters such as bandwidth, delay, reliability, load, and Maximum Transmission Unit (MTU). Of these five parameters, the two most dominant in the calculation of metrics are bandwidth and delay. EIGRP measures the minimum bandwidth on the path from source to destination as well as the accumulated delay on that path. The calculation of these metrics results in a cost value, where the path with the smallest metric value will be selected as the best path. The algorithm used by EIGRP to determine the best path is the Diffusing Update Algorithm (DUAL), a sophisticated algorithm that allows EIGRP to efficiently search and validate paths and detect routing loops. Thanks to DUAL, EIGRP is able to maintain network stability by avoiding ineffective delivery of data packets due to loops.

The performance of EIGRP in choosing the best path can be seen from several aspects, including convergence speed, route stability, flexibility in various network topologies, and efficiency in using network resources such as bandwidth and CPU. In terms of convergence, EIGRP is known to have excellent speed compared to traditional distance vector protocols such as RIP. When a topology change occurs, for example due to a broken link, EIGRP will quickly perform a route update using DUAL without having to send the entire routing table as in RIP. This causes network recovery to run faster and minimise disruption to ongoing network services. EIGRP also excels because it supports a feasible successor mechanism, which is a pre-verified backup path that can be used immediately if the primary path fails. This provides a significant advantage in the context of networks that have many redundant paths, as it can reduce response time to path failures. EIGRP can also perform load balancing on up to six paths with equal or unequal cost load balancing metrics, which allows for a more even distribution of data traffic and thus prevents network congestion.

While EIGRP has many advantages, there are also some limitations when it comes to selecting the best path. One of the main criticisms of EIGRP is that its calculations rely heavily on internal metric parameters and do not explicitly use mathematical approaches such as the graph algorithm used in OSPF (Open Shortest Path First). In some cases, the path selected by EIGRP may not be the most optimal in terms of shortest physical distance or smallest number of hops, but rather the path with the lowest combined metric according to predefined parameters. This leads to the possibility of choosing a technically longer path that is better based on EIGRP's metrics, which may not suit the needs of applications that are highly sensitive to delay or distance travelled [5].Several studies compare the performance of EIGRP with Dijkstra's algorithm used by OSPF. Dijkstra's algorithm mathematically finds the shortest path based on the path weights between nodes, so it theoretically always produces the optimal path in terms of distance. Although EIGRP is more efficient in terms of network overhead and convergence, Dijkstra's approach offers more accurate results in terms of optimal path selection, especially in large networks with many nodes and alternative paths. Therefore, the performance of EIGRP is often considered to need to be tested with a comparator such as Dijkstra to find out how effective and efficient the resulting paths are, and whether there are still optimisation gaps.

EIGRP has advantages in ease of configuration and integration on Cisco devices. In corporate network topologies that use Cisco devices widely, EIGRP is often the first choice due to the full support of the Cisco IOS operating system. In addition, EIGRP also supports the use of VLSM (Variable Length Subnet Mask), automatic summarisation, and fast detection of network topology changes. However, due to its proprietary nature, EIGRP could initially only be used on Cisco devices, although it has now been opened on a limited basis in the form of EIGRP lite for interoperability with other vendors. This becomes a challenge in heterogeneous networks involving various types of devices from different vendors. EIGRP performance assessment can also be seen from simulation results in various network scenarios, such as ring, mesh, or star topologies. In many simulation results, EIGRP shows good performance in terms of average delay, high throughput, and low packet loss rate. However, the effectiveness of path selection still depends on the configuration of metric parameters and network traffic conditions. In high-load networks, EIGRP is still able to balance traffic by utilising multiple paths effectively, but it still cannot match Dijkstra's mathematical precision in determining the absolute shortest path.

EIGRP's performance is also closely related to the network administrator's capacity to develop an optimal configuration. Inaccuracies in metric or summarisation settings can lead to inefficient paths being chosen or even cause routing loops. Therefore, an in-depth understanding of the working mechanism of EIGRP is very important for network managers. In addition, the use of simulation or monitoring tools such as Wireshark, GNS3, and Cisco Packet Tracer can help in evaluating the paths formed and analysing network performance in real time.

3.2. Dijkstra's Algorithm Determines the Shortest Path in the Same Network Topology

Dijkstra's algorithm is one of the most well-known and widely used algorithms in graph theory to determine the shortest path from one source node to all other nodes in a weighted graph that has non-negative weights. It was introduced by Dutch computer scientist Edsger W. Dijkstra in 1956 and became the basis for many applications in computer networking systems, digital mapping, and path planning in navigation systems. In the context of computer networks, Dijkstra's algorithm works by interpreting the network topology as a graph, where each network device (such as a router or switch) is considered a node, and the connections between devices (cables or links) as edges that have weights. These weights can be interpreted as metrics that represent delay, bandwidth, physical distance, or a combination of several parameters that affect the quality of data communication [6]. In determining the shortest path in a network topology, Dijkstra's algorithm works on the greedy principle, by always choosing the path that appears to be the shortest at each iteration before finally reaching the destination node. The process starts by setting one node as the source, then calculating the minimum distance to all other nodes that can be reached directly or indirectly from the source. The algorithm stores the information about the shortest paths known so far and updates it as new vertices are explored. The shortest path found does not necessarily have the least number of hops, but it is the path with the smallest total weight overall, which means it is more optimal in terms of the parameters used as weights.

The basic steps of Dijkstra's algorithm can be explained as follows. First, initialisation is done by setting the distance from the source node to itself as zero, and all other nodes are given an infinite value (∞) as the initial distance. Then, all vertices are put into a set of unvisited vertices. At each step, the vertex with the shortest distance from the unvisited source will be selected. From this vertex, all its neighbours will be evaluated: if the distance from the source to the neighbour through this vertex is smaller than the previous distance, then the new distance will be stored as the temporary shortest path. This process continues until all vertices have been visited or until the destination vertex is found. Finally, once all vertices have been evaluated, the shortest path from the source to the destination can be reconstructed by retracing the travelled vertices.

As an illustration, suppose in a network topology there are five vertices A, B, C, D, and E, with a certain weight between each vertex that represents the delay time. If we want to find the shortest path from node A to node E, Dijkstra's algorithm will evaluate all possible paths from A to E through all other nodes and choose the path with the lowest accumulated weight. For example, A-C-E may be longer in number of hops than A-B-E, but if the delay on paths A-C and C-E is much smaller than A-B and B-E, then Dijkstra's algorithm will choose path A-C-E as the shortest path. Thus, the path selection decision is based on the total weight, not merely the number of hops or topological order.

In real implementations on computer networks, Dijkstra's algorithm is often used by link-state based routing protocols such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS). These protocols rely on a complete map of the network topology stored in each router to calculate the shortest path using Dijkstra. Each router will form a Link-State Database (LSDB) containing complete information about connectivity and weights between nodes in the network. When there is a topology change, such as the addition or failure of a link, the router will update its LSDB and re-run Dijkstra's algorithm to recalculate the best path to all destinations in the network. This allows the routing system to always choose the most efficient path based on the latest topology information.

One of the main advantages of Dijkstra's algorithm is that it is deterministic, i.e. it always produces the same results if the inputs and network conditions are the same. Moreover, it is very efficient in medium to large scale network scenarios, as it only evaluates relevant paths and does not repeat unnecessary calculations. The time complexity of this algorithm depends on the implementation of its data structure; by using a heap (e.g. binary heap or Fibonacci heap), the complexity can be reduced to $O((V + E) \log V)$, where V is the number of vertices and E is the number of edges. In the context of computer networks, this means that the algorithm is well suited for use in routing protocols that require fast but accurate path selection.

Dijkstra's algorithm has several limitations in its application on real networks. One of the main challenges is the need for complete topological information, which means each router must have comprehensive information about the entire network. This increases the network overhead as it requires periodic exchange of routing information. In addition, Dijkstra does not consider dynamic parameters such as real-time changes in traffic load or unstable link conditions, unless regular LSDB updates are performed. As a result, a theoretically optimal path may no longer be efficient if there is a traffic spike or link breakage shortly after the calculation is completed [7].

In comparison to the EIGRP protocol which uses a hybrid method and the DUAL (Diffusing Update Algorithm) algorithm, Dijkstra's algorithm has a different approach. If EIGRP chooses a path based on a combination of metrics designed by the network administrator (e.g. bandwidth and delay), then Dijkstra chooses a path solely based on the smallest total weight from source to destination. Therefore, although EIGRP is very efficient in Cisco environments and has fast convergence capabilities, Dijkstra's algorithm is superior in terms of mathematical accuracy in path selection. This comparison is important in research on routing performance because it can help determine whether the paths selected by protocols such as EIGRP are really close to optimal or can still be improved further by algorithmic approaches.

In network simulations, Dijkstra's algorithm can be implemented using various software such as Cisco Packet Tracer, GNS3, or programming simulations using Python or Java. In these simulations, users can design a network topology with a number of nodes and weights between connections, then run Dijkstra's algorithm to determine the shortest path from one node to another. This result can then be compared with the path chosen by an actual routing protocol, such as EIGRP, to assess whether the routing decisions made by the system are optimal or need to be intervened. The evaluation is done by measuring parameters such as hop count, total delay, and path stability.

By using Dijkstra's algorithm in a simulation of the same network topology used by the EIGRP routing protocol, researchers can gain a more objective understanding of EIGRP's effectiveness in selecting paths. If the path chosen by EIGRP is very different from the path found by Dijkstra, then it can be concluded that the metric configuration on EIGRP still needs to be adjusted to get closer to the optimal path. On the other hand, if both produce similar paths, it can be said that EIGRP has worked efficiently in adjusting routing decisions based on the parameters used. In the latest development of network technology, Dijkstra's algorithm remains the main reference in the development of path optimisation systems, including in the context of Software Defined Networking (SDN) and the Internet of Things (IoT). In an SDN environment, the central controller can use Dijkstra's algorithm to calculate the shortest route globally before disseminating routing information to all network devices. Meanwhile, in IoT systems that have limited resources and connectivity, a lightweight version of Dijkstra's algorithm can be modified to reduce the need for computing power and memory.

3.3. Differences in Path Efficiency Generated between EIGRP and Dijkstra's Algorithm Based on Network Performance Parameters

Comparison of path efficiency generated by Enhanced Interior Gateway Routing Protocol (EIGRP) and Dijkstra's algorithm is important in assessing the effectiveness of routing systems in a computer network. These two approaches have different working principles and path selection methods. EIGRP as a dynamic routing protocol relies on a combined metric to select the best path, while Dijkstra's algorithm uses mathematical principles to determine the path with the smallest weight. Although both are designed to achieve the same goal-that of efficiently delivering data from a source to a destination-the results obtained from each method can vary greatly depending on the network topology structure, traffic conditions, and the performance parameters being referenced. Therefore, it is important to understand how much difference in path efficiency the two methods produce in real and simulated network contexts [8].

EIGRP uses a composite metric that considers several parameters, namely minimum bandwidth, cumulative delay, reliability, load, and Maximum Transmission Unit (MTU). In practice, only two main parameters are used by default in the metric calculation: bandwidth and delay. The result of this calculation will determine the cost of a path, and EIGRP will choose the path with the smallest metric as the best path. On the other hand, Dijkstra's algorithm calculates the shortest path from one node to all other nodes in a weighted graph, based on the smallest weight without explicitly considering network metrics like EIGRP. In other words, EIGRP is more flexible because it can be configured according to the needs of the network, but it can be less mathematically precise than Dijkstra.

In terms of efficiency, the path chosen by EIGRP may be longer (in number of hops) than Dijkstra's if it has higher bandwidth and lower delay. This is because EIGRP's main goal is not to find the path with the least number of hops, but rather the path that is overall more reliable for data delivery based on certainparameters. For example, if there are two paths from source to destination-path A with 2 hops and medium bandwidth, and path B with 3 hops but high bandwidth and low delay-then EIGRP tends to choose path B. Meanwhile, Dijkstra's algorithm will still choose path A if the total weight (e.g. based on physical distance or delay) is smaller. Thus, mathematically Dijkstra provides a more optimal path in terms of total weight, but not necessarily superior in terms of throughput or load balancing.

In various network simulations, the difference in efficiency between EIGRP and Dijkstra's algorithm can be seen through network performance parameters such as delay, throughput, packet loss, and jitter. Simulation results on a fairly complex mesh or star topology show that the path calculated by Dijkstra tends to have a lower average delay than the path chosen by EIGRP. This is due to the deterministic approach used by Dijkstra in choosing the shortest path. However, in terms of throughput and network stability during topology changes, EIGRP shows better performance. This is because EIGRP has a fast recovery mechanism through feasible successor and load balancing, whereas Dijkstra must recalculate the entire path when changes occur.

In addition, Dijkstra does not automatically consider dynamic factors such as link reliability or traffic levels, unless the system is designed to continuously update edge weights based on current conditions. This means that a theoretically optimal path may become less efficient if it experiences traffic spikes or link disruptions. On the other hand, EIGRP can adapt to network conditions in a more real-time manner, as it receives updates from neighbours regularly and recalculates metrics dynamically. In this sense, although Dijkstra's resultant path appears more efficient from a mathematical perspective, EIGRP offers practical efficiency in real network conditions that are always changing.

When viewed from the number of hops, Dijkstra's path results are usually more efficient because always looks for the shortest path based on the smallest weight, while EIGRP does not make hops the main consideration. However, a smaller number of hops does not necessarily mean that the path is better. In some cases, a path with more hops but with larger bandwidth and smaller delay actually results in faster and more stable data delivery performance. Therefore, path efficiency cannot be seen from just one parameter, but must be assessed from a combination of network performance parameters. In simulations conducted using Cisco Packet Tracer or GNS3 with a topology of 6 to 10 routers and several alternative paths, comparison of EIGRP and Dijkstra route results shows that the difference in efficiency ranges from 10-25% depending on network conditions and the parameters measured. For example, under idle conditions with light traffic, Dijkstra can produce paths that provide up to 15% lower delay than EIGRP. However, when traffic spikes or topology changes occur, EIGRP's path can outperform Dijkstra's due to its fast failover capability and multiple path (load balancing) options that Dijkstra's algorithm lacks in its basic version.

One aspect that differentiates the two is their ability to cope with network changes. EIGRP has a very fast convergence time thanks to the use of the DUAL algorithm. This makes EIGRP more responsive when changes occur, such as link breakage or the addition of new nodes. Meanwhile, Dijkstra's algorithm requires a thorough recalculation of all affected paths in the event of a change, making it more slow to adjust. In real-time networks such as VoIP or video streaming, the speed of convergence largely determines the quality of service. In this context, path efficiency is not only seen by how short the path is, but also how fast the system can find and switch to a new path that is feasible.

4. Conclusion

Based on the results of the study and analysis, it can be concluded that the routing protocol EIGRP has superior performance in terms of convergence speed, configuration flexibility, and adaptation to changes in network topology. EIGRP is effective in selecting the best path based on combined metrics such as bandwidth and delay, but in the context of mathematical efficiency of shortest paths, Dijkstra's algorithm proves to be more optimal as it calculates the path with the smallest total weight systematically. Comparison through simulation shows that paths generated by Dijkstra tend to have smaller delay and hop count, while EIGRP is more stable under dynamic network conditions and supports load balancing. Therefore, integration or cross-evaluation between EIGRP and Dijkstra's algorithm can provide a more thorough insight in improving network routing efficiency and reliability. This research confirms the importance of a hybrid approach in the development of routing systems that re more intelligent, adaptive, and suited to the needs of today's network topologies.

References

[1] K. Finata and K. Nasution, "Analisis Kinerja Protokol Routing Open Shortest Path First (OSPF) pada Jaringan Universitas Islam Sumatera Utara," *sudo J. Tek. Inform.*, vol. 3, no. 2, pp. 76–89, 2024, doi: 10.56211/sudo.v3i2.532.

[2] Rizki Indra Saputra, "Analisis Kinerja Redistribution Routing Protokol OSPF, EIGRP dan BGP," pp. 1–115, 2021.

[3] M. A. MF, "ANALISIS KINERJA PROTOKOL OSPF DAN RIPv2MENGGUNAKAN ONOS CONTROLLER PADA JARINGANSOFTWARE DEFINED NETWORKS (SDN)," 2024.

[4] B. Ulummudin and M. Suartana, "Mengatasi Link Failure Pada Software Defined Network Dengan Routing Ulang Menggunakan Algoritma Dijkstra," *J. Informatics Comput. Sci.*, vol. 03, pp. 193–199, 2021.

[5] G. Purnama, "Studi Kinerja Protokol Routing Dinamis Ospf Menggunakan Simulator Opnet," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3s1, pp. 990–996, 2023, doi: 10.23960/jitet.v11i3s1.3431.

[6] S. Binti Dona, D. M. Wiharta, and N. Indra ER, "Analisis Kinerja Protokol Routing Dinamis Is-Is Untuk Meningkatkan Performansi Jaringan Hybrid Kampus Unud Denpasar," *J. SPEKTRUM*, vol. 10, no. 4, p. 348, 2023, doi: 10.24843/spektrum.2023.v10.i04.p40.

[7] S. Arifin, "Kinerja dmvpn phase 3 dan mpls l3 vpn dengan routing protocol ospfv2, eigrp, bgpv4 dan ipsec terhadap layanan video streaming menggunakan gns3," *Repository.Uinjkt.Ac.Id*, 2024, [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/77068%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/77068/1/SAMSUL ARIFIN-FST.pdf

[8] N. Iryani, A. D. Ramadhani, and M. K. Sari, "Analisis Performansi Routing OSPF menggunakan RYU Controller dan POX Controller pada Software Defined Networking," *J. Telekomun. dan Komput.*, vol. 11, no. 1, p. 73, 2021, doi: 10.22441/incomtech.v11i1.10187.