Jurnal Teknologi Informasi dan Pendidikan
Volume 18, No. 1, March 2025
https://doi.org/10.24036/jtip.v18i1.886

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Capacity Management Strategy for Handling Traffic Load with Dijkstra Algorithm at PT XYZ

Najma Syarifa Rahmah[1*✉], Faqih Hamami[1], Riska Yanu Fa'rifah[1]
[1]Information System, Faculty of Industrial and System Engineering, Universitas Telkom, Indonesia
*Corresponding Author: najmasyarifa@student.telkomuniversity.ac.id

Article Information

Article history:

No. 886
Rec. August 03, 2024
Rev. March 04, 2025
Acc. March 05, 2025
Pub. March 19, 2025
Page. 678 – 696

Keywords:

- *Network congestion*
- *Network traffic optimization*
- *Capacity management strategy*
- *Dijkstra Algorithm*
- *Quality of Service*

ABSTRACT

In the digital era marked by rapid data growth, network congestion occurs when network capacity cannot handle the transmitted data volume, becoming a significant challenge that impacts the Quality of Service (QoS). This study proposes an effective capacity management strategy, which includes alternative routing, load balancing, and capacity enhancement, involving the optimized Dijkstra Algorithm and a Decision Support System (DSS) to improve QoS. The Dijkstra Algorithm is effective in reducing the load on congested routes and ensuring balanced traffic distribution, which enhances the capacity and reliability of the network. The results indicate a reduction in packet loss from 142.353 to 66.340 packets (or 53,39%) and an increase in the number of packets successfully transmitted from 122.542 to 198.555 packets (or 61,98%). Specific node capacity increases have significantly bolstered data transmission success. This comprehensive strategy not only improves the reliability of data transmission and network performance consistency but also enables the network to adapt to growing demands without degrading performance, confirming the efficacy of capacity management in addressing capacity challenges and improving QoS.

How to Cite:

Rahmah, N. S., Hamami, F., & Fa'rifah, R. Y. (2025). Capacity Management Strategy for Handling Traffic Load with Dijkstra Algorithm at PT XYZ. Jurnal Teknologi Informasi Dan Pendidikan, 18(1), 678-696. https://doi.org/10.24036/jtip.v18i1.886

1. INTRODUCTION

The continuous advancement of technology over the years is vital for IT companies [1], especially in dealing with the increasingly complex challenges of processing large volumes of data [2]. Limited resources and diverse traffic can impact data transmission, leading to network congestion and reduced service quality [3], [4]. Network congestion severely affects Quality of Service (QoS), a key measure of network performance, as it can cause low throughput, packet loss, transmission inequity, and longer delays [5], [6]. Traffic overload occurs when the network's capacity cannot handle the data volume, potentially harming the network and reducing lifespan [6], [7], [8]. Therefore, capacity plays a crucial role in ensuring efficient data transmission.

The primary factors in managing network capacity are the computing nodes (which cover aspects like power, bandwidth, memory, and storage) and the availability of nodes within the network. Network quality becomes more stable as capable and interconnected nodes rise [9]. Effective and dynamic network capacity management is crucial for controlling operational costs [10].

Masinde et al. [9] proposed a capacity management strategy that includes data storage and replication, routing, and load balancing. Nakagawa et al. [11] suggested an approach for managing capacity that involves using the shortest alternative path for data processing to handle congestion. If all paths are unavailable, the requests are rejected. Shen et al. [12] proposed capacity management that involves expanding network capacity by prioritizing nodes, evaluated using metrics like degree centrality, clustering coefficient, and betweenness centrality.

Networks are commonplace in virtually every company, university, and educational institution, facilitating data transmission within these environments [13]. PT XYZ has also faced network congestion due to excessive capacity utilization, impacting a disruption in service quality and network performance. Consequently, this research proposes a comprehensive approach to capacity management to address the traffic overload issue at PT XYZ. The proposed strategies include alternative routing, load balancing, and capacity enhancement.

Route selection plays a role in this strategy, as balanced data routing helps prevent node overload. Routing requirements include factors such as distance, path load, cost, and travel time, which complicate pathfinding. However, this classic problem in graph theory can be addressed with the Dijkstra Algorithm. This algorithm is renowned for determining the shortest path between two nodes with minimal cost [14], [15], [16], [17], [18]. Furthermore, many researchers have applied this algorithm to pathfinding in various domains, including road traffic, logistics delivery, and IoT (Internet of Things) devices [14], [17], [19], [20].

This research aims to optimize network traffic at PT XYZ, particularly in areas experiencing overload. The expected outcome is to enhance load distribution and improve QoS, thereby reducing the impact and losses faced by the company and improving network quality for long-term use.

2. RESEARCH METHOD

Research methodologies provide a framework for conducting the different stages of research [21]. This paper utilizes the Decision Support System (DSS) framework to create a systematic approach to problem-solving. DSS is a computerized tool that delivers pertinent information to aid decision-making, particularly for semi-structured problems. The stages involved in this research include data preparation, optimization, and measurement.

2.1. Data Preparation

2.1.1. Data Collection

The acquired data includes network topology data from PT XYZ, referred to as *"relationlink,"* which contains 277 rows, and *"request"* data, divided into uplink and downlink data, each with 4,900 rows. Each dataset has *'froma'* and *'tob'* columns indicating the starting and destination points. The *"relationlink"* dataset includes a *'totalcap'* column representing the capacity of each network and a 'cost' column for the usage cost of the routes. Both *"request"* datasets contain columns for service type and bandwidth.

2.1.2. Data Preprocessing

The preprocessing stage is the initial step to prepare the data for use, involving data merging, duplicate removal, handling missing and inconsistent values, data transformation, and feature development. First, uplink and downlink data merged into *"traffic"* data. Then, all datasets are cleansed by removing duplicates. In the *"relationlink"* dataset, missing values in the 'cost' column are imputed with zero. Also, data inconsistency will be resolved by selecting the highest capacity and lowest cost for node pairs with differing values. Data transformation for '*totalcap'* unit to Gbps by multiplying the original values by 1000. Finally, feature engineering involves aggregating the *'totalkap'* column to determine the capacity for each node.

2.2. Optimization

The optimization method involves capacity management strategies, including alternative routing, load balancing, and capacity enhancement, to address congestion caused by overload.

The network topology of PT XYZ uses a weighted, directed graph representing nodes and the paths between them, with costs associated with each path. This graph, denoted as $G = (V, E, L)$, consist a set of nodes V, directed paths E connecting source node u to destination node v, and weights L indicated by path labels. Figure 1 is the graph construction step that involves adding nodes (i) and paths to the graph, with additional attributes such as capacities on nodes and costs on edges (ii), to accurately represent the network and facilitate the analysis of interconnections and network resource management.

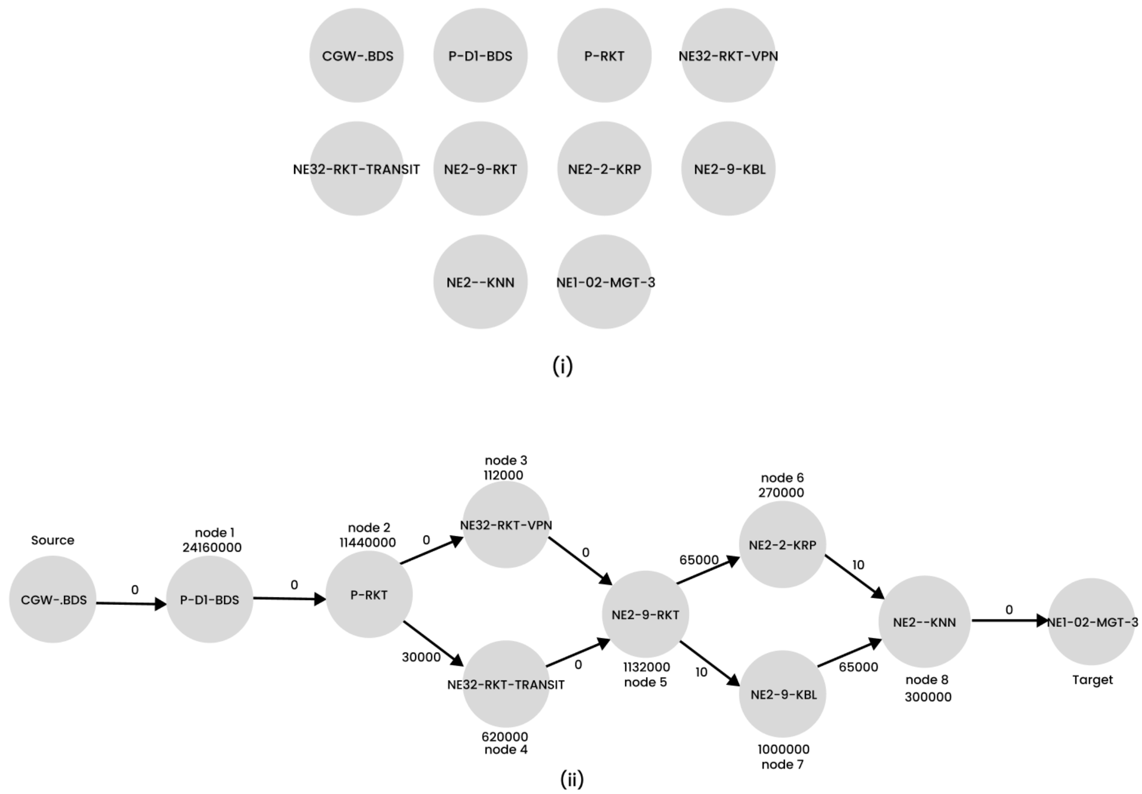


Figure 1. Sample of Graph Construction

2.2.1. Alternative Routing

Routing is crucial for efficient and fast data transmission, optimal resource utilization, and reducing congestion risk. The Dijkstra algorithm optimizes routing by finding the shortest path with the lowest cost. If a path is congested, it seeks alternative routes to maintain transmission. Figure 2 shows the workflow of the Dijkstra algorithm.

https://doi.org/10.24036/jtip.v18i1.886

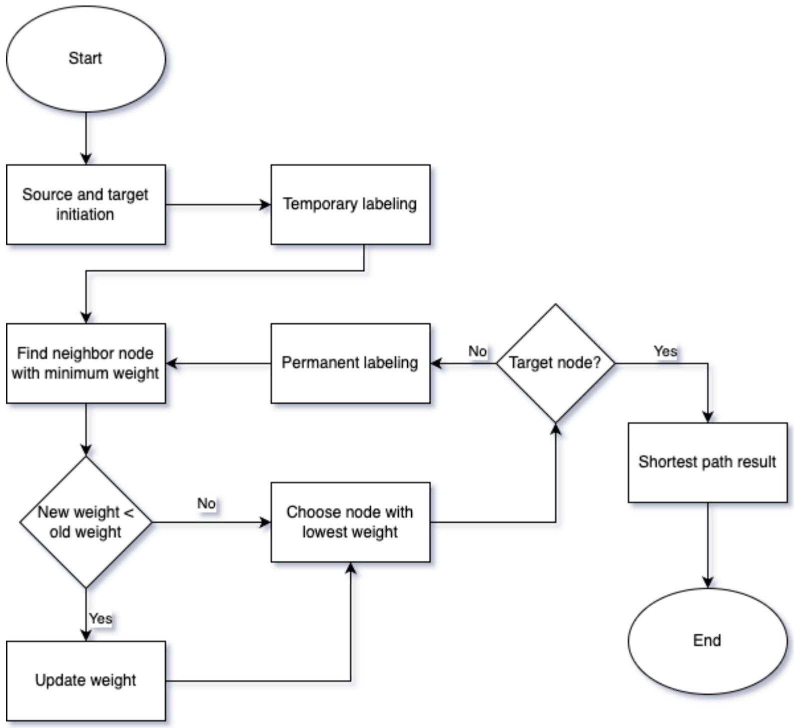


Figure 2. Dijkstra Algorithm Flow

Explanation of the algorithm workflow based on Figure 2:

1. Initialize the start node (u) and the destination node (v).
2. Temporarily label the start node with 0 (zero) and assign ∞ (infinity) to all other nodes.
3. Calculate the weight at each neighboring node.
4. Find the minimum value among all nodes still labeled temporarily.
5. Change the temporary label for the node with the minimum weight to a permanent label. If there is more than one node, choose arbitrarily.
6. Repeat steps 3 through 5 until reaching the destination node.
7. Save the calculation results and the path, then display the results.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

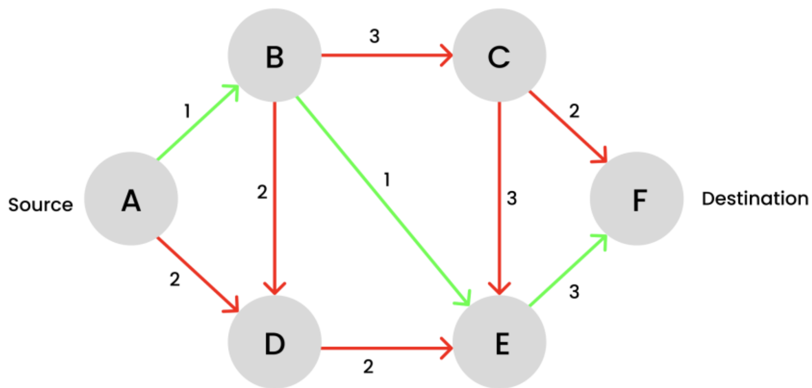


Figure 3. Dijkstra Algorithm Result

2.2.2. Load Balancing

Load balancing occurs by removing nodes with excessive traffic. As shown in Figure 4, congestion occurs after the networks are processing 4.504 requests, and congestion occurs at Node 3 when handling request 4.505 due to insufficient capacity. The transmitted bandwidth is 320, exceeding Node 3's capacity of 179. Therefore, Node 3 is removed, preventing the algorithm from using that path for alternative routes. The green path indicates the shortest actual route, and when congestion is detected, the blue path represents the shorter alternative route for data transmission.

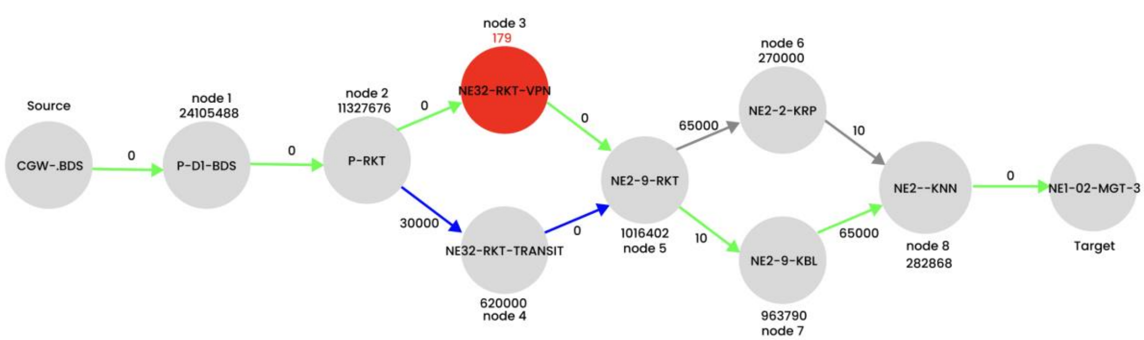


Figure 4. Sample of Alternative Shortest Path Result

2.2.3. Capacity Enhancement

When alternative paths are not available, capacity enhancement becomes the solution by prioritizing nodes based on criteria such as degree centrality, betweenness centrality, clustering, and node utility. If the node is not a priority node, the request will be rejected.

Degree centrality represents the number of connections a node has in a network graph. The centrality formula, defined in (7), where Σ deg(v) represents the total degree of node v.

$$C_d(v) = \Sigma \deg(v) \tag{7}$$

The centrality scores obtained will be normalized using formula (8), where N-1 with N represents the number of nodes in the graph.

$$C_d(v)' = \frac{C_d(v)}{N-1} \tag{8}$$

Betweenness centrality is determined to ascertain how frequently a node acts as an intermediary in the shortest path between pairs of other nodes in the network. Formula (9) defines betweenness centrality calculation. *V* represents the set of all nodes in the graph, where $\sigma(s,t)$ is the number of shortest paths from node s to node t, and $\sigma(s,t|v)$ is the number of shortest paths from node *s* to node *t* that pass-through node *v*.

$$C_b(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{9}$$

This calculation result will normalize to achieve a similar scale using formula (10). (N-1)(N-2) represents the maximum number of shortest paths that can pass through a node in a graph, where N is the total number of nodes.

$$C_b(v)' = \frac{2C_b(v)}{(N-1)(N-2)} \tag{10}$$

Clustering coefficient is determined using formula (11), which considers *T(u)* as the number of directed triangles through node *v*. The directed triangle is a cycle of three interconnected nodes. The notation $deg^{tot}(v)$ indicates the total number of in degree and out degree of node *v*, while $deg^{\leftrightarrow}(v)$ represents the reciprocal degree, i.e., the number of bidirectional paths connecting node *v* to other nodes.

$$C_c(v) = \frac{T(v)}{2(deg^{tot}(v)(deg^{tot}(v)-1) - 2deg^{\leftrightarrow}(v))} \tag{11}$$

The accumulation of node centrality can determined in formula (12).

$$C_n(v) = \frac{C_d(v)' + C_b(v)' + C_c(v)}{3} \tag{12}$$

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

The utility is calculated to assess how effectively the network utilizes its capacity for data transmission according to formula (13), with utility above 100% indicating overload, which affects the selection of priority nodes for accurate capacity enhancement.

$$U(v) = \frac{total\ packet\ size}{maximum\ capacity} x\ 100 \tag{13}$$

Based on the metrics above, the priority nodes are obtained using formula (14).

$$P_n(v) = C_n(v)\ x\ U(v) \tag{14}$$

Bega et al. [10] allocated capacity based on traffic needs over time. Shen et al. [12] expanded capacity by considering expenditure costs and their impact on revenue. Anbaran et al. [22] expanded capacity by accounting for production costs and power consumption. This study limits capacity increases to 100% of the initial capacity for priority nodes to avoid over-expansion and consider data limitations.

2.3. Measurement

Measurements were conducted to evaluate the performance of the optimization method. This analysis assesses the traffic load on each node and calculates Quality of Service (QoS) based on packet delivery and packet loss. The results provide recommendations for the company to improve network quality.

3. RESULTS AND DISCUSSION

3.1 Graph Analysis

This section contains an analysis of graph exploration based on centrality calculation, such as degree centrality, betweenness centrality, and clustering coefficient. It will also explain the priority of nodes to be used as criteria for capacity addition when there is an overload on a node. The graph exploration results appear in Figure 5.

3.1.1. Degree Centrality

Degree centrality measures the connection level node-to-node, indicating its importance in the network. As shown in Figure 5a, the top 10-degree centrality results highlight node NE2-DMO with a value of 1,000, demonstrating its high connectivity in data transmission, followed by nodes with lower connectivity levels.

3.1.2. Betweenness Centrality

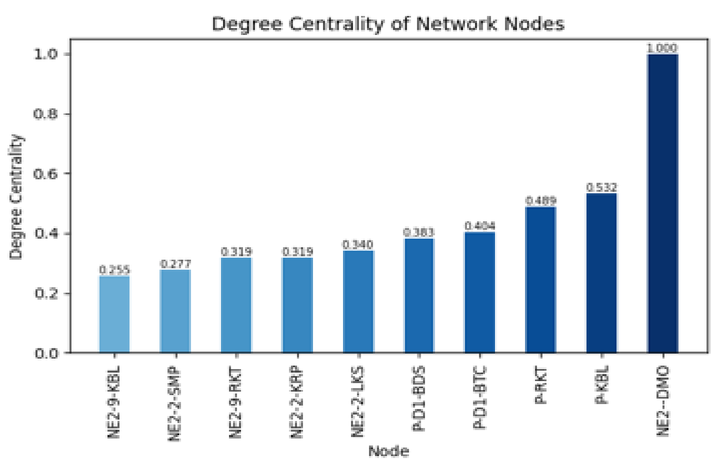
Betweenness centrality measures how often a node appears in the shortest paths between other nodes, acting as a communication intermediary. Figure 5b shows the top 10 betweenness centrality values, with node NE2-9-RKT leading at 1, highlighting its importance in data flow. Other nodes with lower values still play a role in network redundancy and stability.

3.1.3. Clustering Coefficient

The clustering coefficient measures the closeness of node relationships in the network. Figure 5c shows the top 10 nodes with the highest clustering coefficient values, with node NE2-2-LKS leading at 0,075, indicating a cluster of interconnected neighbors. Nodes with lower values, like NE2-9-RKT, may have many neighbors that are less interconnected but still act as intermediaries, as evidenced by NE2-9-RKT's high betweenness centrality.

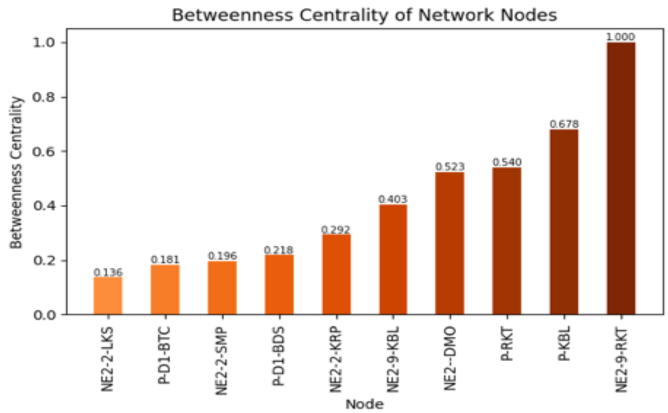
3.1.4. Utility

The utility value measures how efficient capacity utilization is to meet network transmission needs. Figure 5d lists the top 10 nodes with the highest utility values, notably NE2-4-KBL, NE32-RKT-VPN, and NE3-KBL-VPN, all exceeding 100%. Indicates an imbalance between available capacity and demand, leading to network overload.

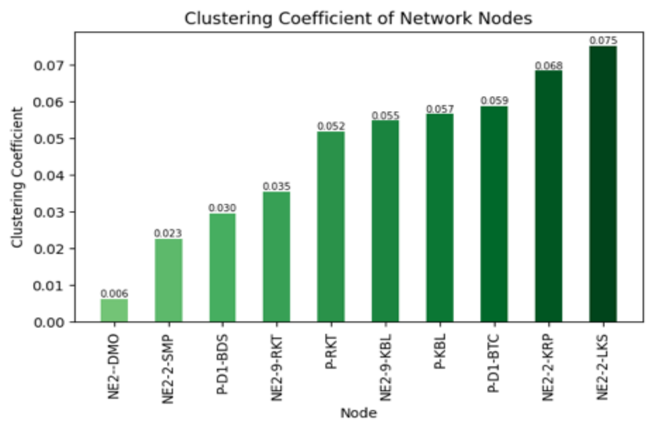


(a)

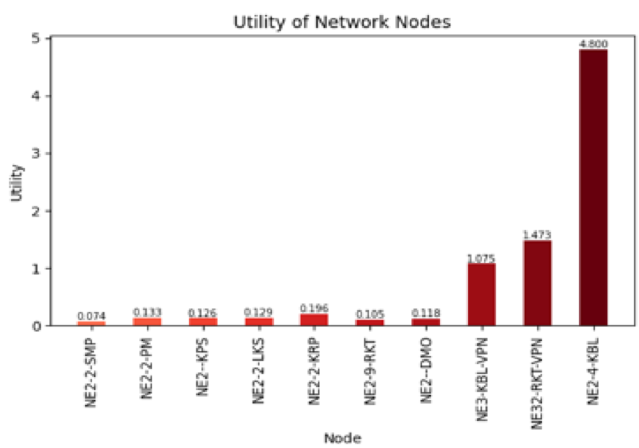
https://doi.org/10.24036/jtip.v18i1.886



(b)



(c)



(d)

Figure 5. Graph Analytics Result, (a) Top 10 Degree Centrality Nodes, (b) Top 10 Degree Centrality Nodes, (c) Top 10 Clustering Coefficient Nodes, (d) Top 10 Utility Calculation

Jurnal Teknologi Informasi dan Pendidikan
Volume 18, No. 1, March 2025
https://doi.org/10.24036/jtip.v18i1.886

3.1.5. Node Priority

Node prioritization is determined based on the cumulative results of the total centrality metric values and utility at each node. This importance is associated with the strong influence of each node in communication and network coherence. The nodes not only serve to connect other nodes but also efficiently facilitate transmission flow commensurate with traffic demand. Nodes with high priority levels become the primary focus for enhancing capacity to uphold network performance.

The results of node priority calculations are illustrated in Figure 6, indicating the top 10 nodes that will have the opportunity for capacity enhancement if alternative paths are unavailable and the remaining capacity cannot accommodate further requests.

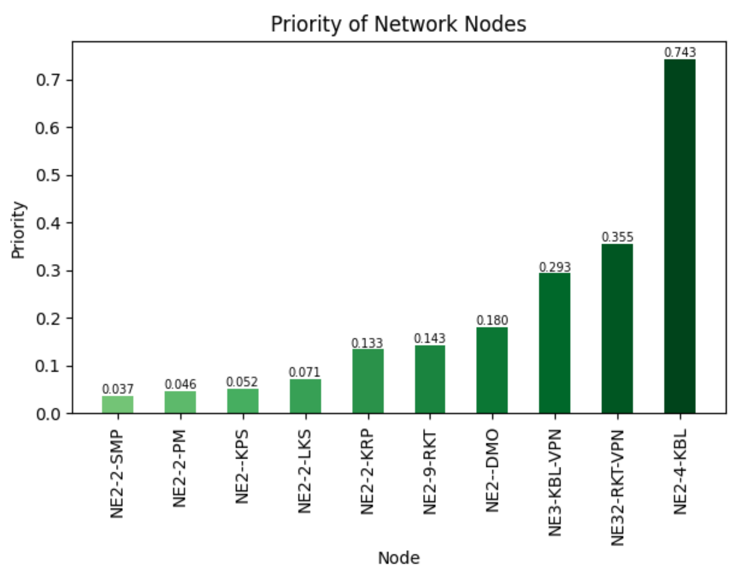
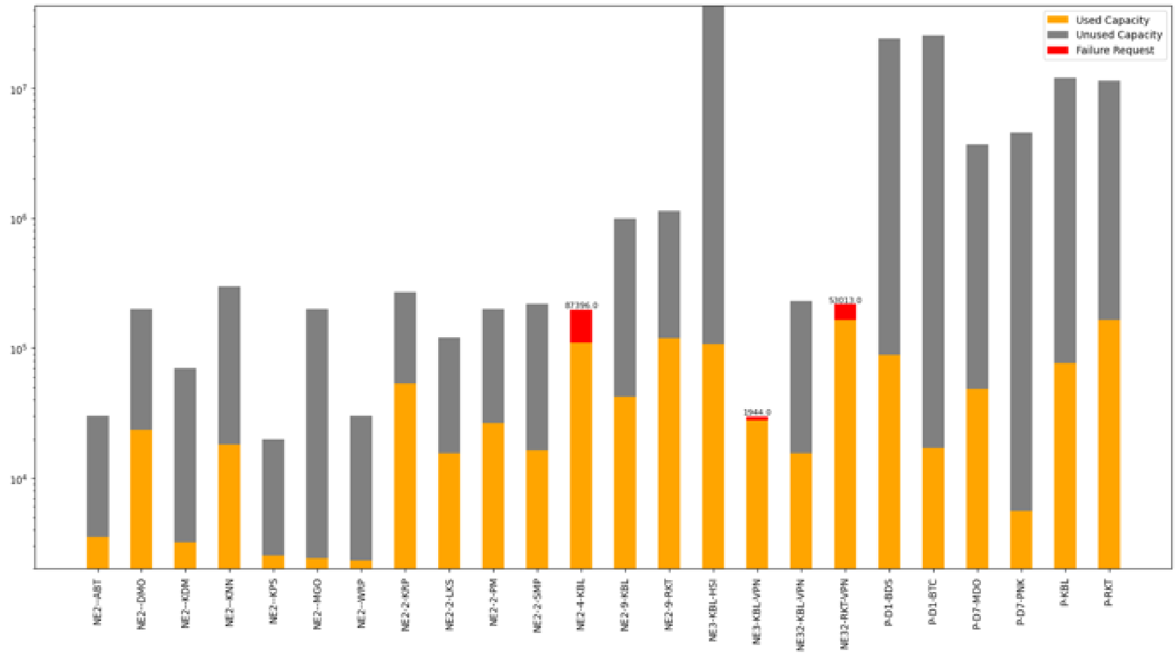


Figure 6. Top 10 Priority Nodes

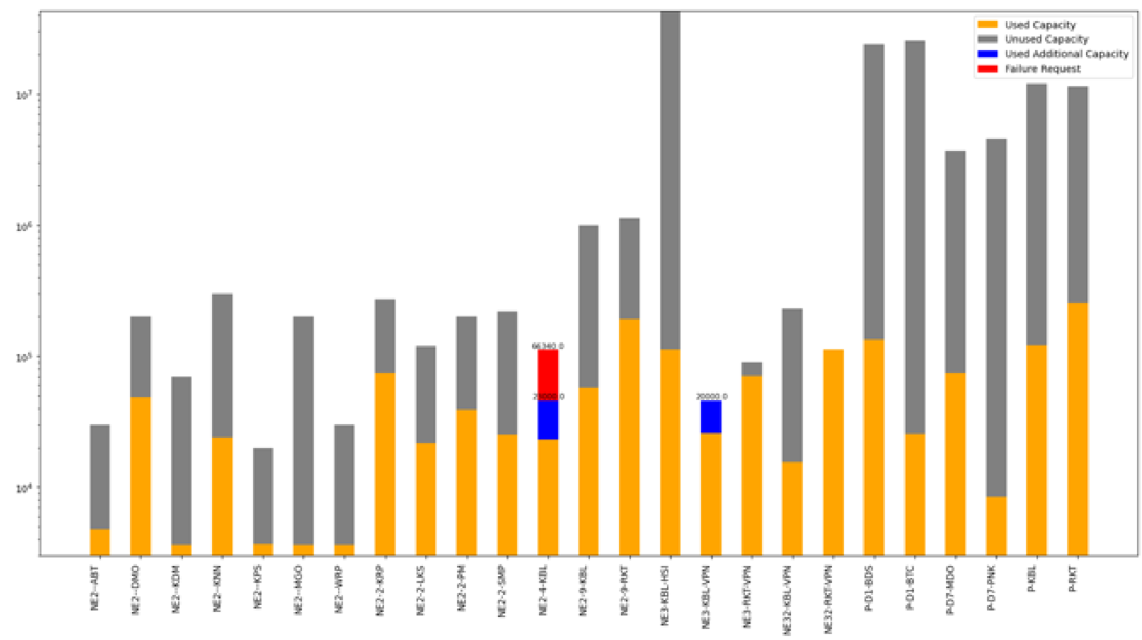
3.2 Comparison of Load Allocation

Based on the results of load allocation calculations for each node through network traffic, Figure 7a and 7b portray the comparison of allocation load for each node before and after optimization.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id



(a)



(b)

Figure 7. Load Allocation Results. (a) Load Allocation Before Optimization, (b) Load Allocation After Optimization

Three of the 24 nodes are congested due to overloaded traffic in Figure 7a, i.e., NE2-4-KBL, NE3-KBL-VPN, and NE32-RKT-VPN. As shown in Table 1, NE2-4-KBL only has 23.000 capacity with high demand and can handle only 20,83% of demand. NE3-KBL-VPN and NE32-RKT-VPN have higher capacity, and success rates raise 93,4% and 67,87%. The fact that even nodes with large capacity can still congest highlights the complexity of the network and the variation of data traffic, indicating that careful management of load allocation is needed to anticipate traffic increase in the future.

Table 1. Load Allocation Details Before Optimization

Node	Capacity	Total Demand	Success Rate
NE2-4-KBL	230000	110396	20,83%
NE3-KBL-VPN	26000	27944	93,04%
NE32-RKT-VPN	112000	165013	67,87%

However, the optimization results using the proposed capacity management strategy in Figure 7b show a reduction of overloaded nodes during the data transmission process. While Figure 8a shows 24 nodes, Figure 8b shows 25 nodes, which means it has additional nodes, i.e., NE3-RKT-VPN, to reallocate the load of the congested nodes.

Table 2 details the load reallocation of three congested nodes, which have an increased success rate for each node. NE3-KBL-VPN and NE32-RKT-VPN successfully reallocated the load to other nodes and raised 100% both for each success rate with a capacity enhancement of 20.000 for NE3-KBL-VPN, while NE32-RKT-VPN did not need any enhancement. In contrast, node NE2-4-KBL still cannot transmit the entire request despite the 100% capacity increase, while the enhancement still has positive results of increasing the success rate.

Table 2. Load Allocation Details After Optimization

Node	Capacity	Additional Capacity	Total Demand	Success Rate
NE2-4-KBL	23000	23000	112340	40,9%
NE3-KBL-VPN	26000	20000	46000	100%
NE32-RKT-VPN	112000	0	112000	100%

This result reaffirms that the Dijkstra Algorithm can effectively balance the allocation of failed transmission load due to traffic overload, supported by additional capacities at priority nodes to enhance network reliability. Careful load allocation can efficiently manage resource utilization without the need for excessive additional capacity to achieve higher success rates. In the case of NE2-4-KBL, companies should reallocate existing resources to handle excessive demand imbalances.

3.3 Quality of Service Measurement

Network quality is assessed based on two QoS parameters, i.e., packets delivered rate and packet loss rate. The packets delivered represent the number of packets successfully transmitted to the destination [23]. Formula (15) defines the calculation of packets delivered, which involves computing the number of packets received by the destination node divided by the number of packets sent [24].

$$PDR = \frac{packets\ received}{packets\ sent} x\ 100 \qquad (15)$$

Meanwhile, packet loss signifies the total packets lost during the transmission process due to congestion as defined in formula (16), where packets sent minus packets received are the number of packet losses and divided by packets sent to calculate the packet loss rate[23] [25].

$$PLR = \frac{(packets\ sent -\ packets\ received)}{packets\ sent} x\ 100 \qquad (15)$$

PLR are standardized into several categories based on Tiphon [26].

Table 3. Packet Loss Category

Category	PLR (%)
Perfect	0 (zero)
Good	3%
Medium	15%
Poor	25%

3.3.1. Packets Delivered

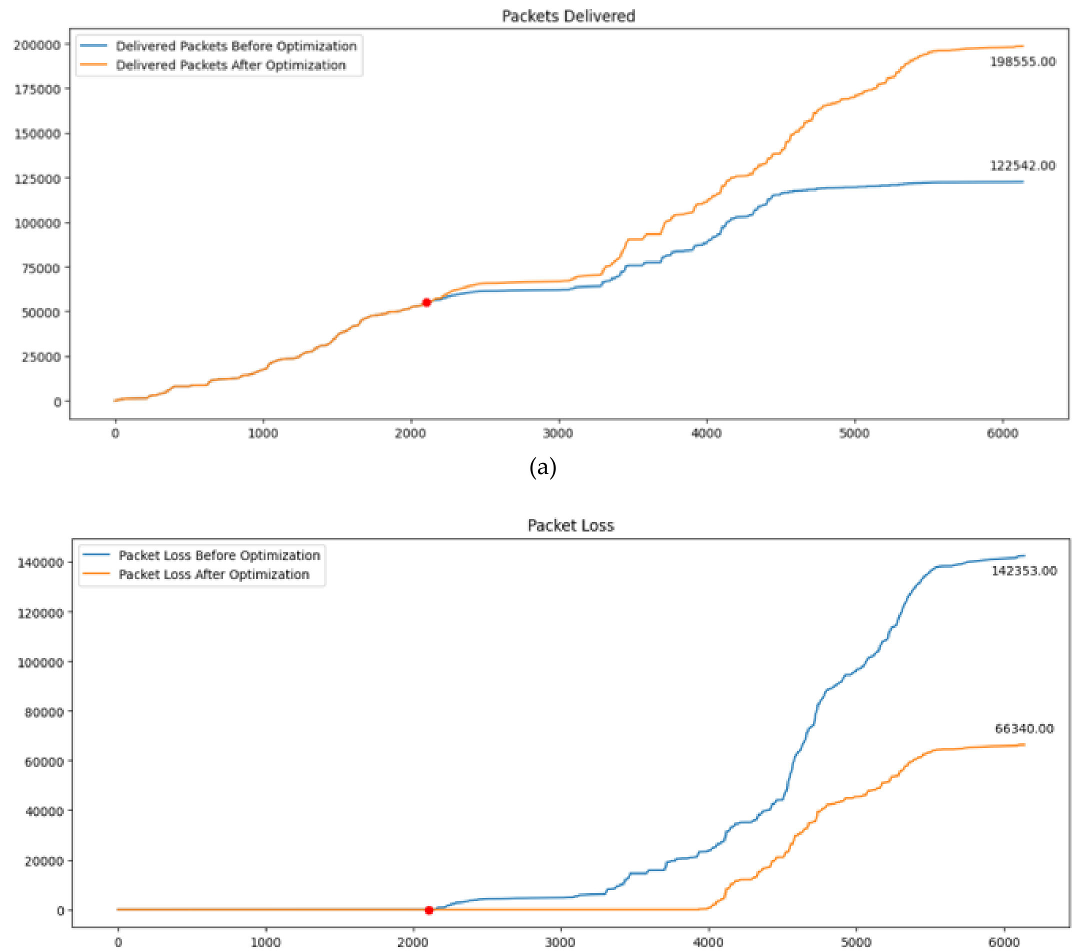
Packets delivered results are shown in Figure 8a. The number of packets delivered, depicted by a blue-colored curve, is 122.542 packets out of 264.895 packets that should be sent, so the PDR result is 46,26%. The optimization results raise the packets delivered value into 198.555 packets, so the PDR result becomes 74,95%. Significant is the increase in the PDR as the efficient capacity management strategy to handle the traffic load up to 61,89% increased.

3.3.2. Packet Losses

In proportion to the increase in packets sent, the decrease in packet loss results show in Figure 8b. The number of packet losses depicted in the blue curve is 142.353, so the PLR result is 53,73%, which is very far below the poor category on Tiphon. The optimization results increase the value of packets sent, depicted in an orange-colored curve where 66340 packets losses and the PLR result is 25,04%, yet still in a poor category. However, capacity

Jurnal Teknologi Informasi dan Pendidikan
Volume 18, No. 1, March 2025
https://doi.org/10.24036/jtip.v18i1.886

management strategy still efficiently decreases the PLR significantly in dealing with network congestion up to 53,39%.



(a)

(b)

Figure 8. QoS Measurement Results, (a) Comparison of packets delivered, (b) Comparison of packet losses

3.4 Comparison of Total Success and Failure Request Results

The optimization results in Figure 9a demonstrate a significant increase in the number of successfully sent requests for each service type. *'ebis'* service experienced an increase of nearly 38%, with successful request numbers rising from 812 to 1,118. A higher surge occurred in '*mobile*' services, reaching approximately 51%, where successful requests increased from 608 to 920. *'Retail'* and *'wholesale'* services also saw significant increases, around 35.4% and 50.3% respectively.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Jurnal Teknologi Informasi dan Pendidikan
Volume 18, No. 1, March 2025
https://doi.org/10.24036/jtip.v18i1.886

Alongside the overall increase in successfully delivered requests, there was a significant decrease in the number of failed requests after the optimization process, as seen in Figure 9b. The *'ebis'* service failures decreased from 574 to 268 failed requests, with a reduction ratio of 53.31%, and *'mobile'* services experienced a similar decrease from 580 to 268, with a reduction ratio of 53.79%. A higher surge occurred in *'retail'* services, with a reduction from 819 to 268 failed requests, representing a reduction ratio of 67.28%. Similarly, *'wholesale'* services failures decreased from 576 to 268, with a reduction ratio of 53.47%.

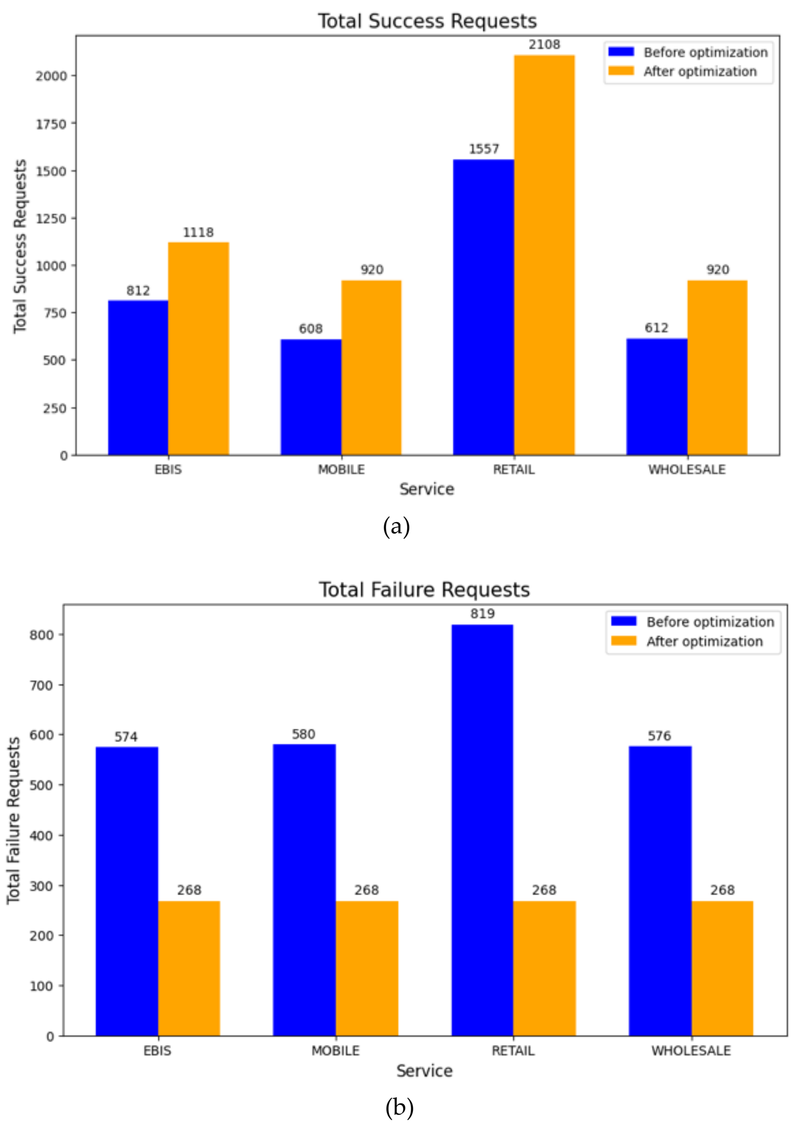


Figure 9. Total Success and Failure Request Calculation Results, (a) Comparison of Total Success Requests, (b) Comparison of Total Failure Requests

Interestingly, after optimization, the number of failed requests sent for each service converged to the same number, i.e., 268. It indicates the effectiveness of the optimization process and the normalization of service quality standards, as well as consistency in handling requests achieved through capacity management strategies, including the utilization of Dijkstra's Algorithm to find alternative routes and capacity enhancements.

4. CONCLUSION

The complexity of technological challenges, such as managing data volume and diverse traffic, underscores the critical role of effective network capacity management. Capacity management strategies that include optimized routing, load balancing, and capacity enhancement are pivotal in alleviating congestion and enhancing Quality of Service (QoS). At PT XYZ, network congestion prompted comprehensive capacity management, resulting in notable efficiency gains. For instance, nodes experiencing congestion achieved a 100% success rate post-optimization, with a significant packet loss reduction of 53,39% and a 61,89% increase in packets delivered, ultimately leading to an improved QoS.

Since the proposed strategy was tested only on PT XYZ's network topology using uplink and downlink data, this study has several limitations. It may not fully represent other network environments with different architectures or dynamic traffic patterns, especially during spikes. In addition to this, it presumes that the condition of the network is reasonably stable and does not take into consideration any real-time variations that can influence performance. This research did not conduct a comprehensive analysis of the cost factors, which included the costs associated with the adoption of new devices or changes to the infrastructure. Moreover, as the findings are based on simulations rather than real-world implementation, performance may vary in live environments.

Despite the enhancements, there are still issues with maximizing network resources and satisfying future demand because of changing traffic patterns. In order to improve capacity management strategy, future research might be focused on an adaptive optimization approach concerning more real traffic situations. Discovering more diverse traffic scenarios and utilizing real-time network monitoring data can enhance strategies for reducing congestion and managing capacity more effectively. Testing the proposed methodologies in various network situations and conducting extensive cost-effective studies will also provide for a better understanding of long-term scalability, practical implementation, and overall efficiency for sustainable network development.

REFERENCES

[1] I. Andhika, “Analysis of Mobile Banking Acceptance in Indonesia using Extended TAM (Technology Acceptance Model),” *Jurnal Teknologi Informasi dan Pendidikan*, vol. 16, no. 2, 2023, doi: 10.24036/jtip.v16i2.626.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

[2] J. Tan *et al.*, "An efficient information maximization based adaptive congestion control scheme in wireless sensor network," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2915385.

[3] M. H. Homaei, F. Soleimani, S. Shamshirband, A. Mosavi, N. Nabipour, and A. R. Varkonyi-Koczy, "An Enhanced Distributed Congestion Control Method for Classical 6LowPAN Protocols Using Fuzzy Decision System," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2968524.

[4] Y. Guo, Z. Mi, Y. Yang, H. Ma, and M. S. Obaidat, "Efficient Network Resource Preallocation on Demand in Multitenant Cloud Systems," *IEEE Syst J*, vol. 13, no. 4, 2019, doi: 10.1109/JSYST.2019.2910276.

[5] K. Kritikos and D. Plexousakis, "Requirements for QoS-based Web service description and discovery," in *IEEE Transactions on Services Computing*, 2009. doi: 10.1109/TSC.2009.26.

[6] Z. Hu, X. Wang, and Y. Bie, "Game Theory Based Congestion Control for Routing in Wireless Sensor Networks," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3097942.

[7] Y. Lu, R. Pan, B. Prabhakar, D. Bergamasco, V. Alaria, and A. Baldini, "Congestion control in networks with no congestion drops," in *44th Annual Allerton Conference on Communication, Control, and Computing 2006*, 2006.

[8] J. Wu, Y. Peng, M. Song, M. Cui, and L. Zhang, "Link congestion prediction using machine learning for software-defined-network data plane," in *CITS 2019 - Proceeding of the 2019 International Conference on Computer, Information and Telecommunication Systems*, 2019. doi: 10.1109/CITS.2019.8862098.

[9] N. Masinde, S. Bischoff, and K. Graffi, "Capacity Management Protocol for a Structured P2P-based Online Social Network," in *2020 7th International Conference on Social Network Analysis, Management and Security, SNAMS 2020*, 2020. doi: 10.1109/SNAMS52053.2020.9336563.

[10] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "AZTEC: Anticipatory Capacity Allocation for Zero-Touch Network Slicing," in *Proceedings - IEEE INFOCOM*, 2020. doi: 10.1109/INFOCOM41043.2020.9155299.

[11] M. Nakagawa, H. Kawahara, K. Masumoto, T. Matsuda, and K. Matsumura, "Performance Evaluation of Multi-Band Optical Networks Employing Distance-Adaptive Resource Allocation," in *25th Opto-Electronics and Communications Conference, OECC 2020*, 2020. doi: 10.1109/OECC48412.2020.9273660.

[12] Y. Shen, X. Wen, L. Wang, W. Guan, and Z. Lu, "Physical Resource Management Based on Complex Network Theory in 5G Network Slice Trading," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2019. doi: 10.1109/WCNC.2019.8885610.

[13] P. Panyahuti, "Validity Test of Motion Graphics Media as A Wide Area Network Learning Media," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 15, no. 1, 2022, doi: 10.24036/jtip.v14i3.572.

[14] A. Alyasin, E. I. Abbas, and S. D. Hasan, "An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method," in *4th Scientific International Conference Najaf, SICN 2019*, 2019. doi: 10.1109/SICN47020.2019.9019345.

[15] A. Buzachis, A. Celesti, A. Galletta, J. Wan, and M. Fazio, "Evaluating an Application Aware Distributed Dijkstra Shortest Path Algorithm in Hybrid Cloud/Edge Environments," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 2, 2022, doi: 10.1109/TSUSC.2021.3071476.

[16] S. Sundarraj, R. V. K. Reddy, M. B. Basam, G. H. Lokesh, F. Flammini, and R. Natarajan, "Route Planning for an Autonomous Robotic Vehicle Employing a Weight-Controlled Particle Swarm-Optimized Dijkstra Algorithm," *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3302698.

[17] K. Wei, Y. Gao, W. Zhang, and S. Lin, "A Modified Dijkstra's Algorithm for Solving the Problem of Finding the Maximum Load Path," in *2019 IEEE 2nd International Conference on Information and Computer Technologies, ICICT 2019*, 2019. doi: 10.1109/INFOCT.2019.8711024.

[18] D. D. Zhu and J. Q. Sun, "A New Algorithm Based on Dijkstra for Vehicle Path Planning Considering Intersection Attribute," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3053169.

[19] H. Y. Chang, P. F. Wang, H. C. Chen, Y. Z. Chen, and D. R. Chen, "On the Study of Shortest-path Problem on Coal-transportation Networks using Dijkstra's Algorithm," in *2019 IEEE International Conference on Consumer Electronics - Taiwan, ICCE-TW 2019*, 2019. doi: 10.1109/ICCE-TW46550.2019.8992003.

[20] S. Kim, H. Jin, M. Seo, and D. Har, "Optimal Path Planning of Automated Guided Vehicle using Dijkstra Algorithm under Dynamic Conditions," in *2019 7th International Conference on Robot Intelligence Technology and Applications, RiTA 2019*, 2019. doi: 10.1109/RITAPP.2019.8932804.

[21] F. P. Kamarudin, "Optimizing Classification Algorithms Using Soft Voting: A Case Study on Soil Fertility Dataset," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 16, no. 2, pp. 255–268, Dec. 2023, doi: 10.24036/jtip.v16i2.800.

[22] A. G. Anbaran, A. Mohammadi, and A. Abdipour, "Capacity enhancement of Ad Hoc networks using a new single-RF compact beamforming scheme," *IEEE Trans Antennas Propag*, vol. 63, no. 11, 2015, doi: 10.1109/TAP.2015.2469092.

[23] M. Purwahid and J. Triloka, "Analisis Quality of Service (QOS) Jaringan Internet Untuk Mendukung Rencana Strategis Infrastruktur Jaringan Komputer Di SMK N I Sukadana," 2019.

[24] M. F. Khan, E. A. Felemban, S. Qaisar, and S. Ali, "Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in wireless sensor networks (WSNs)," in *Proceedings - IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2013*, 2013. doi: 10.1109/MSN.2013.74.

[25] A. Kurnia Saleh, H. Peni Agustin Tjahyaningtijas, and L. Rakhmawati, "Quality of Service (QoS) Comparative Analysis of Wireless Network," 2022.

[26] ETSI, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)," *Etsi Tr 101 329 V2.1.1*, vol. 1, 2020.