مجلة جامعة بني وليد للعلوم الإنسانية والتطبيقية
تصدر عن جامعة بني وليد – ليبيا
Website: https://jhas-bwu.com/index.php/bwjhas/index
المجلد التاسع، العدد الثاني 2024

التوجيه المدرك للسياق

سناء الهادي عبدالكريم ابوشوفة[1]، زهره عبدالله الاشعل[2]
[1] قسم الحاسب الآلي، كلية التربية، جامعة طرابلس، قصر بن غشير، ليبيا.
[2] قسم هندسة البرمجيات، كلية تقنية المعلومات، جامعة طرابلس، طرابلس، ليبيا.
Sa.Abushofa@uot.edu.ly

Context-Aware Routing

Sanaa H. Abdulkarim Abushofa[1], Zahra Abdalla Elashaal[2]
[1] Computer Department, Education College, University of Tripoli, Gaser Ben Gashir, Libya.
[2]Software Engineering, Faculty of Information Technology, University of Tripoli, Tripoli, Libya.

تاريخ الاستلام:06-05-2024   تاريخ القبول: 20-05-2024   تاريخ النشر : 09-06-2024

الملخص:

الطلب المتزايد على تطبيقات الوسائط المتعددة في الوقت الفعلي مثل تدفق الفيديو والصوت عبر الإنترنت يتطلب تركيزًا أقوى على جودة الخدمة (QoS). غالبًا ما تعطي خوارزميات التوجيه التقليدية، مثل أقصر مسار أولاً، الأولوية لأقصر طريق دون مراعاة ازدحام الشبكة أو أنواع حركة المرور. يمكن أن يؤدي ذلك إلى تأخير التطبيقات الحساسة للتأخير مثل مكالمات الفيديو. يظهر التوجيه المدرك للسياق كحل، حيث يقوم بتكييف قرارات التوجيه ديناميكيًا بناءً على نوع الحزمة وظروف الشبكة. من خلال مراعاة عوامل جودة الخدمة، يسهل التوجيه المدرك للسياق تسليم الحزم بكفاءة، مع إعطاء الأولوية للحزم الحساسة للتأخير لتحسين تجربة المستخدم وإنتاجية الشبكة. علاوة على ذلك، تم نقل الحزم بكفاءة، وتم تحقيق المزيد من الإنتاجية من خلال إعطاء الأولوية لتأخير الحزم الحساسة.

الكلمات الدالة: التوجيه المدرك للسياق، الخوارزمية الجينية، أقصر مسار، التأخير، جودة الخدمة*(QoS)* ، الحزم.

Abstract

The growing demand for real-time multimedia applications like video and audio streaming over the internet necessitates a stronger emphasis on Quality of Service (QoS). Conventional routing algorithms, such as shortest path first, often prioritize the shortest route without considering network congestion or traffic types. This can lead to delays for delay-sensitive applications like video calls. Context-aware routing emerges as a solution, dynamically adapting routing decisions based on packet type and network conditions. By considering QoS factors, context-aware routing facilitates efficient packet delivery, prioritizing delay-sensitive packets for improved user experience and network productivity. Moreover, the packets were transferred efficiently, and more productivity was achieved by giving precedence to delay sensitive packets.

Keywords: Context-aware Routing, Genetic Algorithm, shortest path, delay, Quality of Service (QoS), packets.

1. Introduction

Multimedia applications like Skype and Windows Live Messenger use time-sensitive and bandwidth-greedy video and audio content, making them unsuitable for real-time traffic [14]. The current internet is a best-effort network, resulting in unpredictable packet delivery. If a link fails, routers must reroute packets, leading to packet loss, disorder, routing loops, or duplicate packets. To address these issues, techniques like ATM and MPLS networks have been proposed, but they can be time-consuming, expensive, and computationally intensive. The shortest path first method has been ignored, causing problems with current routing methods. Congested links can cause delays in packets, even with high bandwidth. Congestions can be caused by conflicts between packets from different sources passing through the same link. To address this issue, research is being conducted on Context-Aware Routing methods for efficient routing. Overall, the evolving internet technology is causing significant challenges in the realm of multimedia applications. The Open Shortest Path First (OSPF) routing protocol uses the Shortest Path Routing (SPR) method to select the best path between routers and internetwork networks [15]. However, this method relies on Dijkstra and Bellman-Ford algorithms, which can lead to conflicts and bottlenecks, causing delays in packet transfer and potentially affecting multimedia applications [1]. To address this, research is being conducted on context-aware routing, where different agents use a common infrastructure to consider other agents' plans in path planning, avoiding deadlock and congestion, resulting in faster packet delivery and increased productivity [2]. This "Context-aware Routing" study aims to analyze current shortest path routing algorithms and simulate a network that addresses these limitations by selecting the best path based on payload content and avoiding congested links. Objectives include reviewing algorithms, proposing a dynamic algorithm, designing a GUI using Java programming, simulating network diagrams, and analyzing statistics.

Literature Review

Dynamic Routing Protocols are a method that automatically synchronizes routing tables with neighboring routers. These protocols define rules to select the best path to a destination among multiple solutions [5]. There are three main classes of dynamic routing protocols: Distance Vector, Link State, and Hybrid. Distance Vector uses the distance of the destination as a metric to select the best path, with routes with less hops being selected. Link State Routing, also known as shortest path first protocols, calculates the cost of links using metrics like speed, throughput, and bandwidth [6]. Hybrid routing protocols combine characteristics of both distance vector and link state routing protocols, such as EIGRP. These protocols help determine the best path for a network. In today's digital age, real-time multimedia relies heavily on QoS for effective communication. Current routing protocols fail to consider congestion and other node activities, affecting delay-sensitive applications. Context-aware routing is proposed, considering the context of packets, and determining the best path selection, ensuring congestion-free links and optimal bandwidth. This intelligent routing can react differently in different situations [7]. Context-aware routing is a method of planning routes that avoids collisions between packets, resulting in higher throughput. This concept has been extensively researched in the field of Automated Guided Vehicle (AGV) routing, as it is crucial for airports to avoid head-on conflicts [8]. Two main approaches to context-aware routing are the shortest path method and combining both. Early research focused on shortest path selection and alternative routes, while later algorithms like Taghaboni-Dutta and Tanchoco developed efficient algorithms that calculate the delay to pass through resources [2].

2. Context-aware Algorithm

The Genetic Algorithm is a common optimization algorithm used to find the optimum path for packet routing. It uses selection and evolution principles to produce multiple solutions. The algorithm uses chromosomes to create a population randomly, with the fittest chromosome at the top. As time passes, these chromosomes may mutate or crossover to create the next generation. This process continues until a suitable solution is found. The Genetic Algorithm is also used in source routing, where a routing node must know about all nodes in the entire path to the destination. Alternate routes are generated using genetic algorithm operators [13].

2.1 Context-aware using Genetic Algorithm

Source routing, also known as context-aware using genetic algorithms, is a method used to solve path selection problems in routing. It involves routing nodes knowing about all nodes in the entire path to the destination, creating alternate routes using genetic algorithm operators like mutation and crossover. The best route is selected based on its fitness function, with the fitter the route, the higher the probability of selection [13]. Chromosomes are encoded lists of routes from source to destination in a network topology. The shortest path first method is used to create an entry into a routing table, which contains route information useful for the algorithm to analyze route performance. This information is used to apply genetic operators to generate new chromosomes. The best chromosome is selected based on its fitness, which depends on packet transmission delays, which calculated using the equation 1:

$$Packet\ Delay = \frac{size\ of\ the\ packet}{Bandwidth} \qquad \text{Equation 1}$$

The fitness function is calculated using equation 2:

$$f_i = \frac{1/r_i}{\sum_{i=1}^{n} 1/r_i}\ ;\ r_i = {d_i}\Big/{\sum_{i=1}^{n} d_i} \qquad \text{Equation 2}$$

where, d=transmission delay and r=probability of a particular delay.

The chromosome or route with the least delay has the highest fitness value, making it the optimum solution.

Genetic Operators:

Genetic operators are used to generate more chromosomes to select the fittest of all. There are two main types: crossover and mutation. Crossover operations involve replacing intermediate nodes between two routes from the same source and destination with a common node. For example, two chromosomes, V1 and V2, have the same source and destination but different transitional nodes, as shown in Figure 1. New offspring like V'1 and V'2 are generated by swapping parent nodes after a common node, generating two offspring and new routes to be added to the routing table.

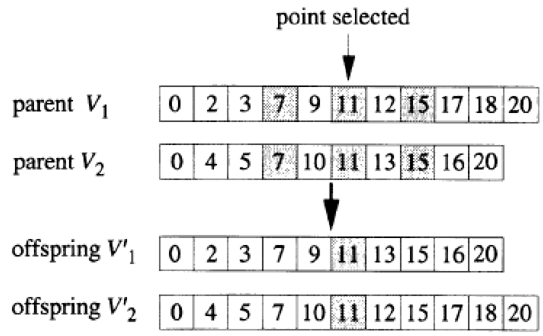

Figure 1 Crossover Operation

Mutation operators generate alternative chromosomes from a chromosome [13]. A node is randomly selected from a mutation node, and a sub route is generated from source 0 using the shortest path method. As shown in Figure 2 a new child or offspring is generated by joining these sub routes, which could already be available in the routing table. However, this offspring is rejected as it could cause loops in a network. Both crossover and mutation procedures are essential for generating the most suitable chromosomes for a network.

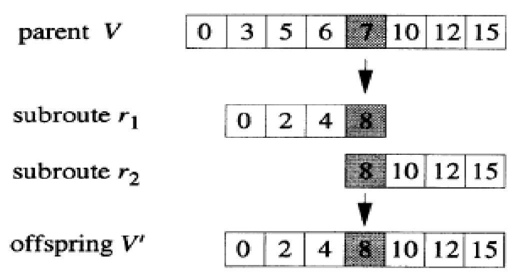

Figure 2 Mutation Operation

The Final algorithm:

The proposed context-aware routing algorithm uses the Genetic Algorithm to route packets intelligently using all available resources. It runs on every node in the network and is invoked when a packet is received. The algorithm checks if the packet is sent by a neighboring node or originated at the same router. If the packet originated at the same router, it populates the routing table with the shortest path and calculates fitness for each route. If the packet is delay sensitive, two best chromosomes are selected based on their fitness, and their child is added to the routing table. The process continues until a set limit is set, and the size of the routing table is maintained. If the number of iterates reaches a set limit, chromosomes with least fitness are removed. The algorithm then checks the frequencies of the routes and sends the packet to the fittest route with the least frequency. If the packet is not delay sensitive, fitness is found from all available shortest path routes and sent using the fittest shortest path.

```
if (Packet received from neighbouring router)
{
        if (current router is destination)
        { Receive file;   }
        else
        { Forward it to the next hop of its route;  }
else if (Packet originated at current router)
```

```
            if (Routing Table is empty)
              {   Populate the routing table with the available solutions;  }
            if (Routing Table has routes)
              {
                  if (Packet == delay sensitive(audio or video))
                   {
                         While (no. of iterates generated < maximum limit)
                            {
                              calculate the fitness of each route;
                              select the two fittest Chromosome and apply Crossover Operator;
                              add the offsprings to the routing table;
                              select a fittest chromosome and apply mutation Operator;
                              add the offspring to the routing table;
                              if (routing table is full)
                                 { Delete route with minimum fitness;  }
                            }
                        for (maximum to minimum fitness route)
                              {   Send Packet to fittest with minimum frequency;
                                        Increase the frequency of the route by 1;
                              }
                     }
                     else if (Packet == simple data)
                          {
                              Send packet through the fittest shortest path;
                          }
            }
    }
```

The basic working of the algorithm is shown as a flow diagram in Figure 3.

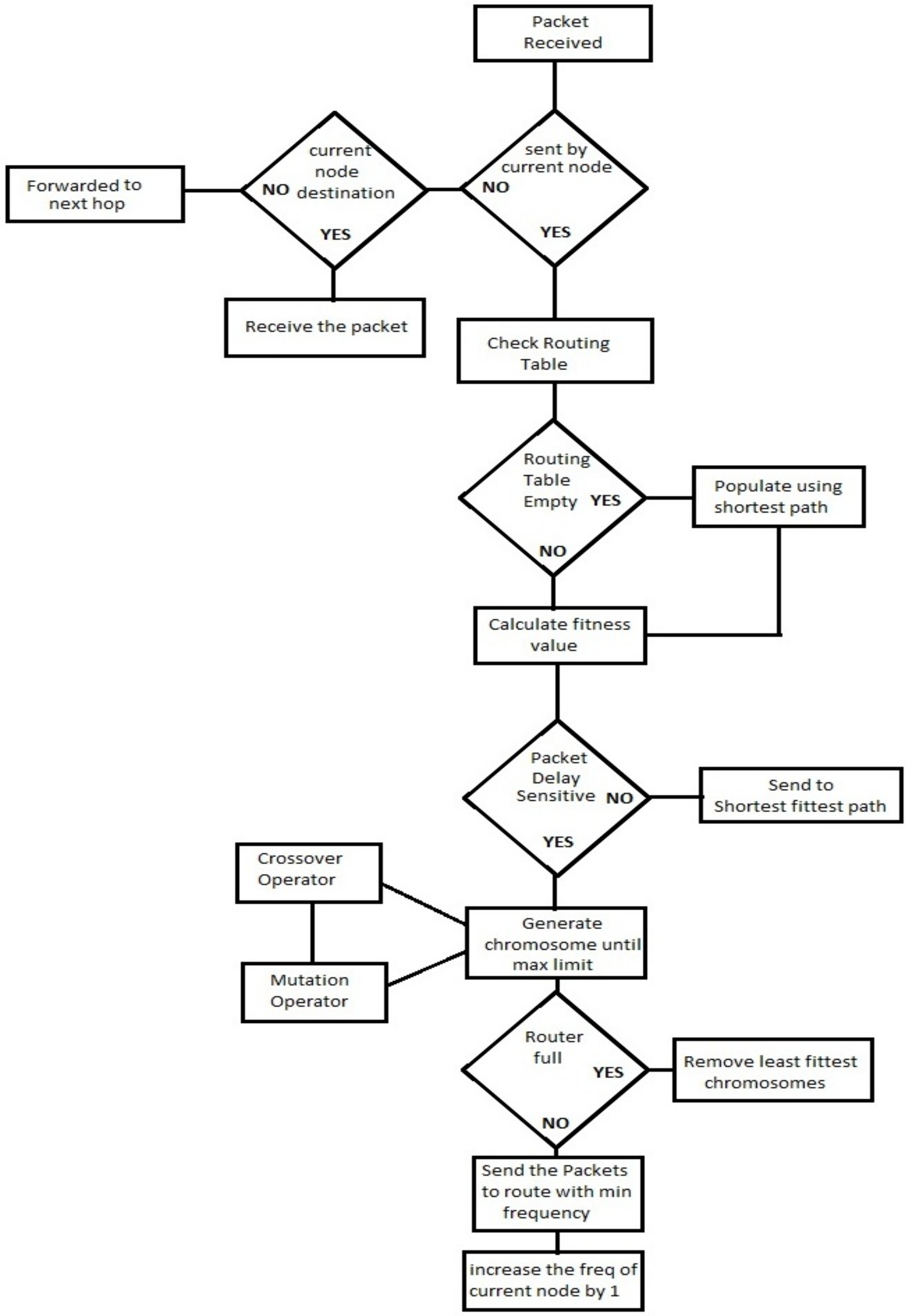

Figure 3 Context-aware Flow diagram.

2.2 Application of Context-aware algorithm

Figure 4 illustrates a network topology with five nodes, with Router 1 as the source and Router 5 as the destination. There are three different links from Router 2 to the destination Router 5, each with different bandwidths and hop counts. The source R1 has three types of files to send to the destination: Video (7Mb), Audio (3Mb), and Plain Text (500Kb).

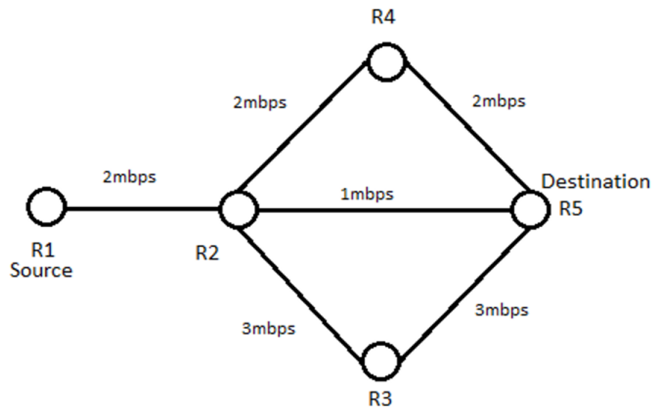

Figure 4 Link Network Topology

To compare the normal shortest path method of Routing with the proposed Context-aware Routing method, the following calculations were used:

1. **Shortest Path Routing Method**: The shortest path from source R1 to destination R5 will be directly from R2 to R5, ignoring longer paths through R4 or R3. Although routes through R3 and R4 are faster with high bandwidth than a direct path, the shortest path will still be used according to Dijkstra Algorithm, the results of the shortest path routing shows in Table 1.

Table 1 Results of the shortest path Routing Method.

	Video file	Audio file	Plain Text file
no. of packets	27803	11915	1935
1st packet delay	0.002112 seconds	0.002112 seconds	0.002112 seconds
delay of rest of packets	58.7178 seconds	25.162368 seconds	4.084608 seconds
Total Delay	58.719936seconds	25.16448 seconds	4.08672 seconds

2. **Context-aware Routing Method**: The context-aware routing method intelligently and quickly transfers packets. It examines the whole network and makes decisions wisely. When the packet arrives at Router 2, it realizes it has longer but high bandwidth links to the destination, which can produce more productivity. It immediately calculates delays and makes decisions accordingly. Table 2 shows the results of the Context-aware routing scenario.

Table 2 Results of the Context-aware Routing Method.

	Video file	Audio file	Plain Text file
no. of packets	27803	11915	1935
1st packet delay	0.001408 seconds	0.002112 seconds	0.002112 seconds
delay of rest of packets	19.572608 seconds	12.581184 seconds	4.084608 seconds
Total Delay	19.574016 seconds	12.583306 seconds	4.08672 seconds

Video is transferred through a high bandwidth link through Router 3, while audio is sent through a relatively higher speed link through Router 4. Plain text files follow the same shortest path.

The results show a dramatic change in delays, as packets were transferred quickly without congestion. This shows that the shortest path is not always good for sending data, and there are more aspects to learn before making a decision for route selection.

3. Methodology and Network Design

The study aimed to conduct an experiment to prove the hypothesis and algorithms. Three software options were considered: OPNET Modeller 14.5, Network Simulator 2 (NS2), and Java. OPNET Modeller 14.5 is a powerful network simulator with extensive features and a smart GUI. However, the project required modifications to the protocol, and NS2 was not feasible due to limited Linux knowledge. Java, a simple, object-oriented language, is used due to its robust nature and security features. It provides encapsulation, inheritance, and polymorphism, which bind code and data together, support hierarchical classification, and reduce complexity by designing generic interfaces for specific actions.

Figure 5 shows Java GUI was designed to implement various topologies and investigate context-aware algorithms. It includes routing nodes capable of intelligently routing packets based on available resources. The GUI can be modified to suit different topologies and network designs.

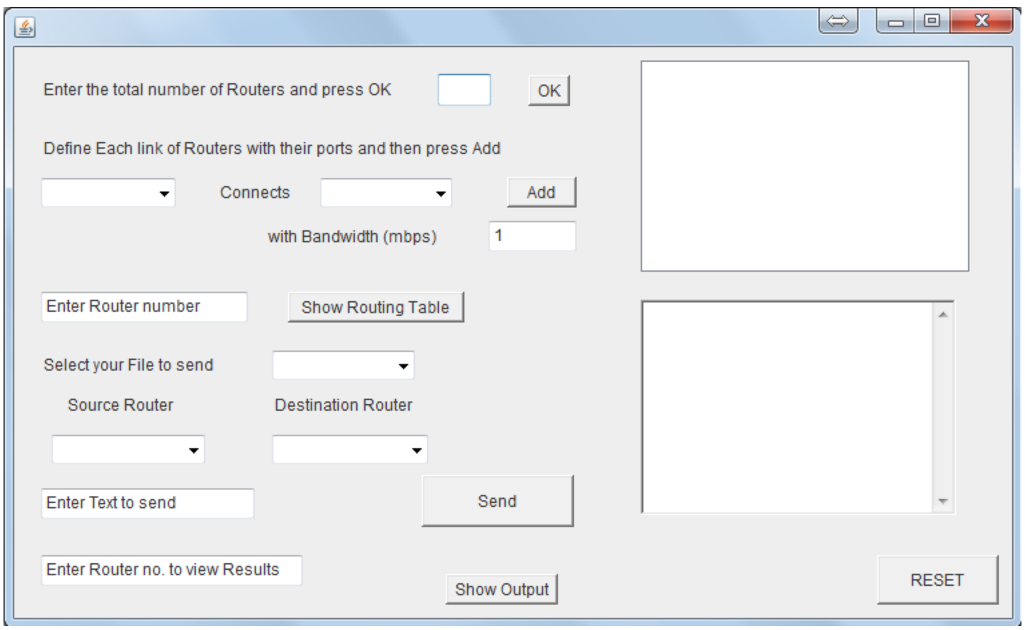

Figure 5 GUI

The study investigates network design, as shown in Figure 6, using a Java GUI and a proposed algorithm to determine the best path for packet routing. Router 1 sends three different files to Router 5, with three different routes diverging from Router 2. The context-aware algorithm determines the optimal router for each type of packet. The shortest path method directs all packets to Router 5 from Router 2, but the proposed context-aware algorithm makes clever decisions based on bandwidth.

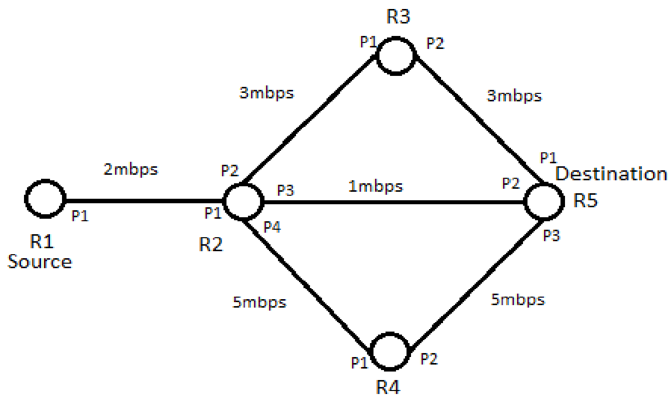

Figure 6 Network Topology

The network is defining the number of routers and their port numbers and bandwidths, and the study assumes different file sizes and types. Figure 7 shows the routing tables for all the routers attached to our network.

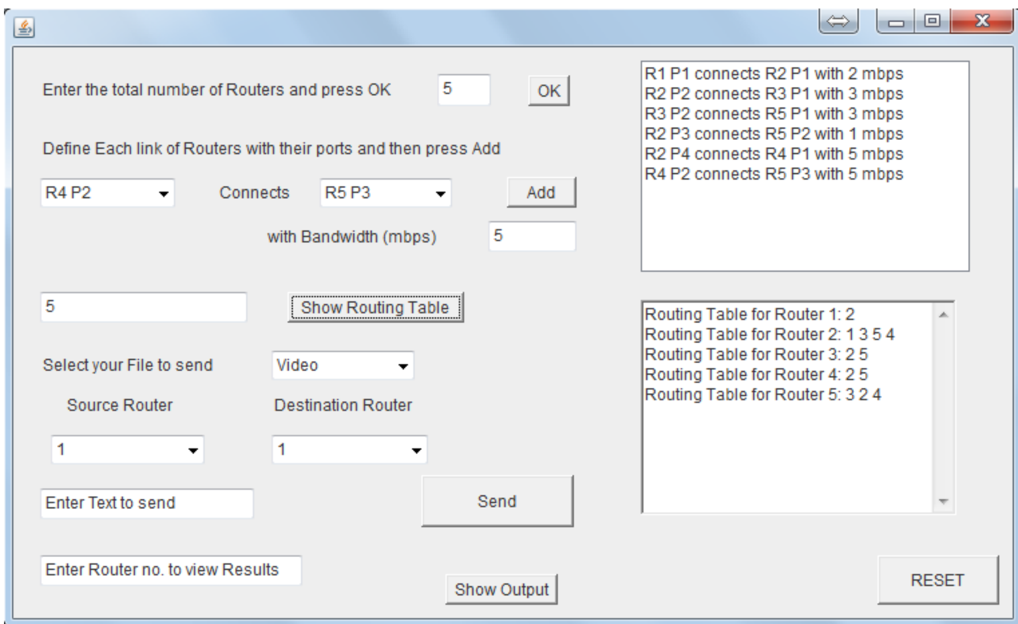

Figure 7 Routing Tables

4. Results and Analysis

The study compares the shortest path algorithm and the context-aware algorithm in transferring files from source R1 to destination R5. The proposed algorithm is used to select the optimum path, unlike the conventional shortest path method. The experimental work demonstrates that if the conventional shortest path method is used, all traffic from R2 is forwarded through a straight link, resulting in collisions, delays, and jitter. This can negatively affect the quality of video and audio files, which are sensitive to time. The shortest path with less bandwidth was chosen, as shown in Figure 8, highlighting the importance of considering all available resources and the context of the packet before making any route decision. The study emphasizes the need for efficient resource utilization and consideration of the packet's context when making route decisions. The results of the study highlight the potential benefits of context-aware routing in network communication.

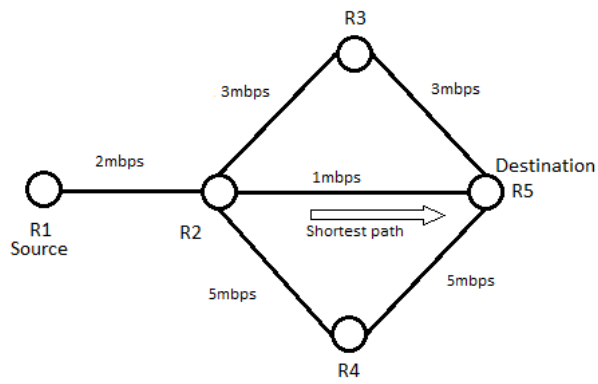

Figure 8 Shortest Path Selection

The context-aware routing method is a network approach that uses intelligent routing to select the best path to a destination. It considers the packet's context and the bandwidth of the link, ensuring that all packets arrive at the destination simultaneously. This method is different from the shortest path method, which forwards all packets to the straight path to Router 5 and results in collisions and delays. In this case, when a video packet arrived at Router 2, it was sent to Router 4 with a 5mbps bandwidth, as it is indicated in Figure 9. The context-aware algorithm chose this route, considering delays as its fitness function. After carrying mutation and crossover operations, it assigned fitness values to all available paths. The path through Router 4 had the highest bandwidth, resulting in the least delays for the video packets. Similarly, when an audio packet was received at Router 2, the context-aware algorithm chose Router 3 with a 3 mbps bandwidth. The algorithm intelligently utilized available resources and directed the audio packets to a free link, avoiding collisions.

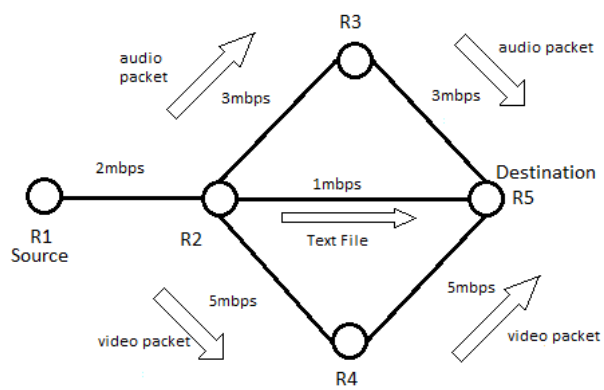

Figure 9 Context-aware Routing algorithm working.

The context-aware routing method also selects the shortest path for text files, as they are not time sensitive and do not cause collisions. This method allows for more efficient use of available resources and reduces the time it takes to transfer all three types of packets.

The context-aware algorithm's performance will be analyzed through graphic calculations and comparisons with shortest path methods, focusing on Packet Delays, Throughput, and Resource Utilization parameters.

4.1 Packet Delay Analysis

Packet delay analysis considers four types of delays: processing, queueing, transmission, and propagation. Transmission delay is the time taken to send bits into the link [10,11], as shown in Equation 3. To compare shortest path and context-aware routing methods, transmission delays for each file are calculated separately.

$$Delay_{Tx} = \frac{no.of\ bits}{link\ Bandwidth} \times total\ packets \qquad \textit{Equation 3}$$

The study focuses on the performance of different routing methods for video and audio files. In the shortest path method, video files are sent directly to Router 5 using the shortest path method. However, using the context-aware routing method, video file packets are forwarded through an indirect link towards Router 4, making full use of high bandwidth available. This results in fewer delays than the shortest path first method, which is the main objective.

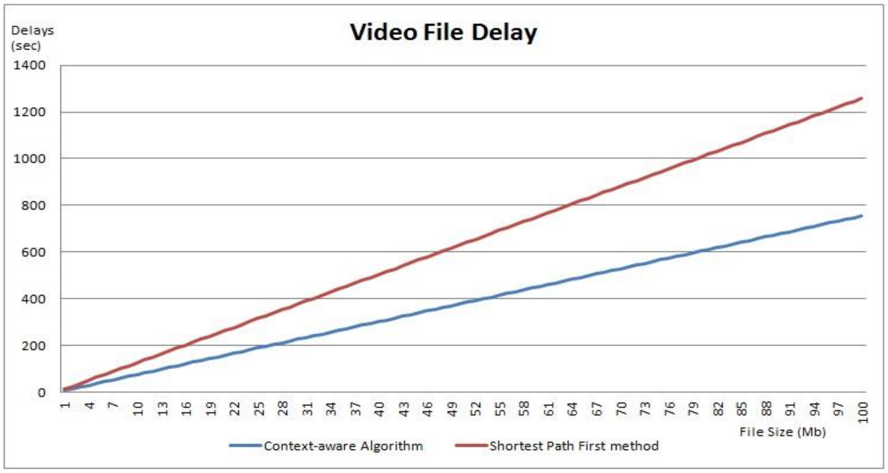

Figure 10 Video File Transmission Delay

Figure 10 shows that the context-aware algorithm causes fewer delays than the shortest path first method, even when taking a longer path. As the file size increases, the context-aware algorithm proves more efficient, as there is a large gap between the values of both methods at a file size of 100Mb.

The context-aware algorithm also performs better when sending audio files, as in Figure 11. When sending a 100mb audio file using the shortest path method, it takes 1258 seconds, while with the context-aware algorithm, all packets are delivered in 978 seconds, saving a lot of time. This shows that packets had to take a longer path but were delivered earlier, and there is less chance of collisions when using separate links to transfer different files.

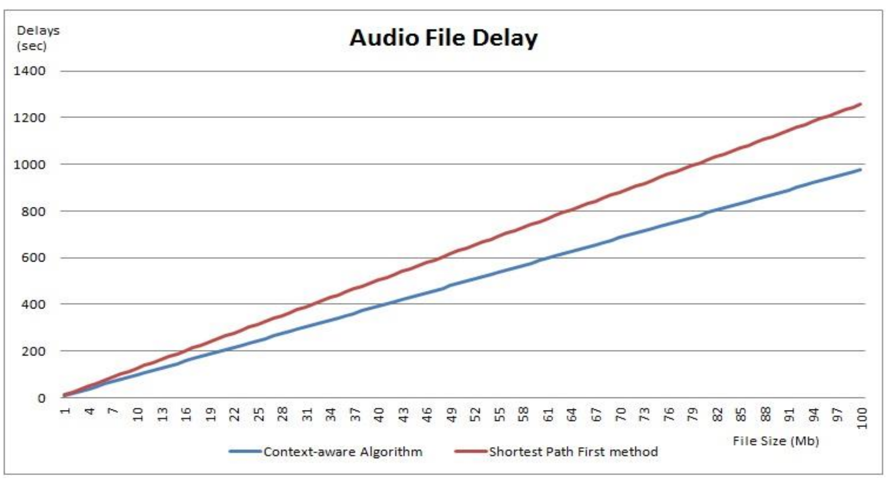

Figure 11 Audio File Transmission Delay

When sending text files, the context-aware algorithm and shortest path first method both have similar delays. Figure 12 for text files shows similar delays using both methods, but the context-aware algorithm is a better choice due to the direct path in the context-aware algorithm taking only text file packets, while the shortest path method carries all packets. Therefore, there will be more delays due to collisions between different packets traveling in the shortest path method.

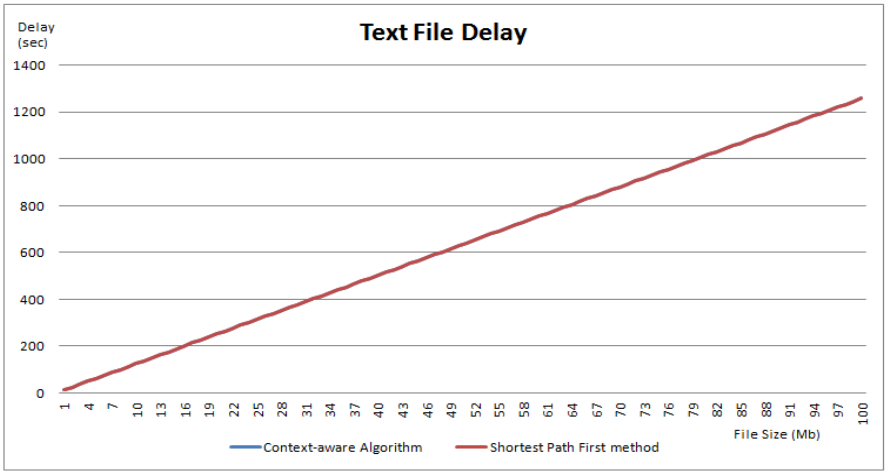

Figure 12 Text File Transmission Delay

The study compared the overall delays caused to send three files, video, audio, and text, using both the shortest path and context-aware routing methods, as cleared in Figure 13. The results showed significant differences between the two methods. In the shortest path method, all packets were transferred using the same link, resulting in the sum of all packet delays. In contrast, the context-aware algorithm separates packets, avoiding collisions and ensuring simultaneous arrival at the destination. The text file took the slowest path, causing the maximum delay.

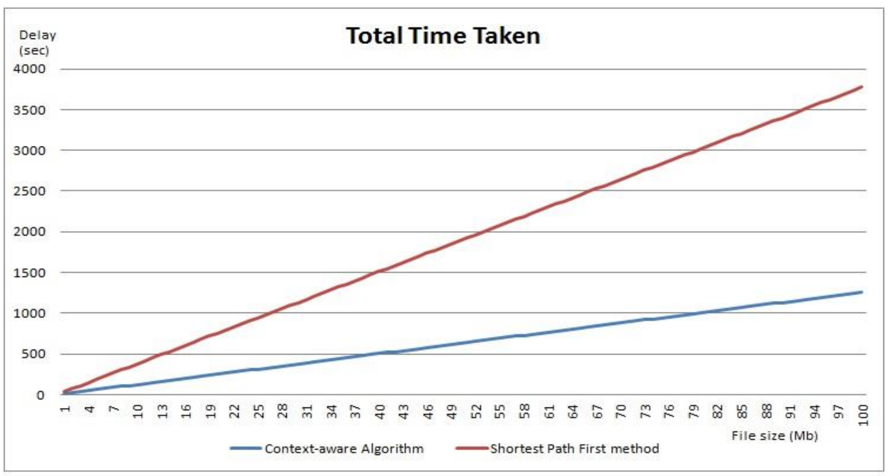

Figure 13 Overall Delays

4.2 Throughput Analysis

Throughput analysis is a mathematical method used to determine the number of packets delivered to the destination successfully over time [12]. It is used to distinguish between different routing methods, such as context-aware and shortest path. Equation 4 was used to calculate the number of packets received by sending different files (Video, Audio, and Text File) using both methods.

$$Throughput = \frac{Total\ no.of\ packets\ (bits) received}{Time\ Taken} \qquad \textit{Equation 4}$$

Video File Throughput, shown in Figure 14, was calculated for both context-aware and shortest path methods. The context-aware algorithm received more packets over time, as it took a longer path to transmit the packets through Router 4 with a link bandwidth of 5mbps. The shortest path direct path to Router 5 was selected using 1mbps link bandwidth.

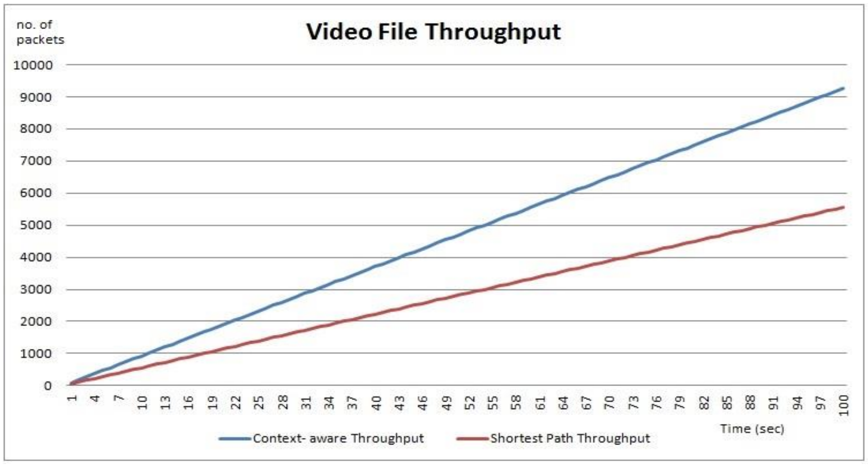

Figure 14 Video File Throughput

Audio File Throughput was calculated for both context-aware and shortest path methods, with the proposed algorithm opting for a longer but faster path, resulting in higher throughput, as shown in Figure 15. The gap between the two methods grew more as time passed, indicating the effectiveness of the context-aware algorithm.

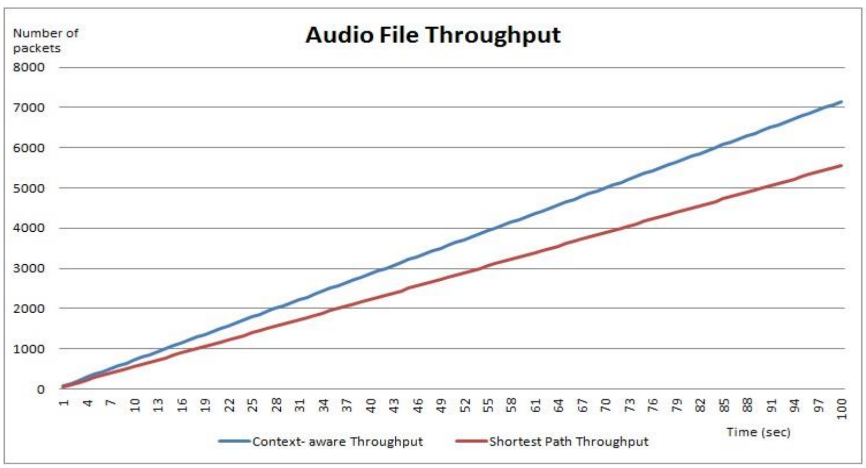

Figure 15 Audio File Throughput

Text File Throughput, in Figure 16, was calculated using both context-aware and shortest path methods with the overall throughput calculated for both types of routing methods.

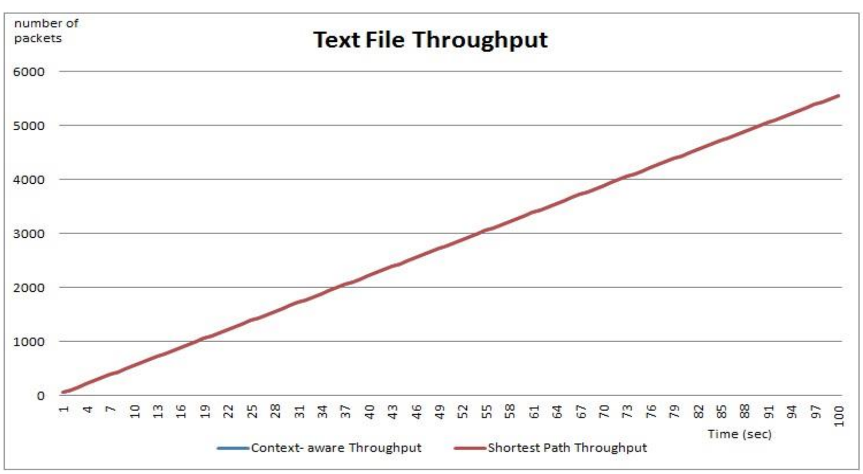

Figure 16 Text File Throughput

The overall throughput of both routing methods was calculated and plotted in Figure 17. The context-aware algorithm received a larger number of packets than the shortest path first method, demonstrating its proficiency in using available resources by sending packets from all available links simultaneously.

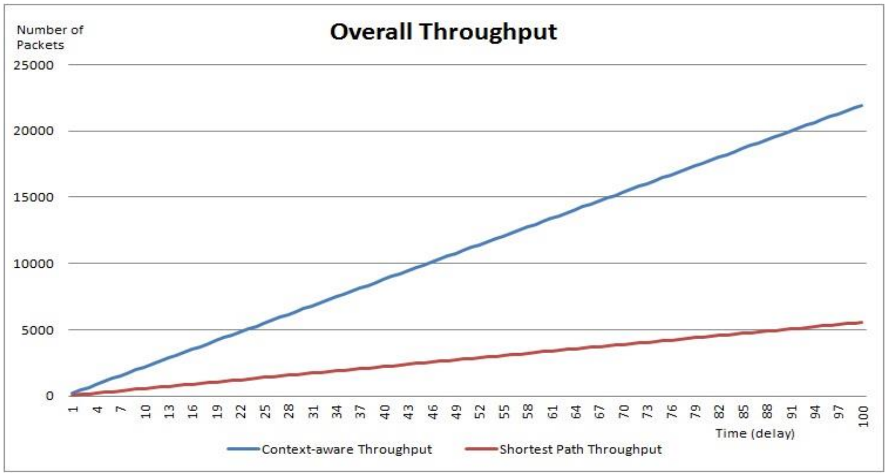

Figure 17 Overall Throughput

4.3 Resource Utilization Analysis

The study analyzes network utilization by comparing context-aware and shortest path methods. The study considers different file types and sizes, such as video, audio, and text files. The data is sent using both methods, and throughput statistics are used to determine resource utilization [16].

The context-aware algorithm used all available resources, transferring all three files simultaneously using different paths, as indicated in Figure 18. The video file took a higher bandwidth path through Router 4, the audio file took a route through Router 3, and the text file took a direct path from Router 2 to Router 5. This resulted in a quick delivery time of only 53 seconds.

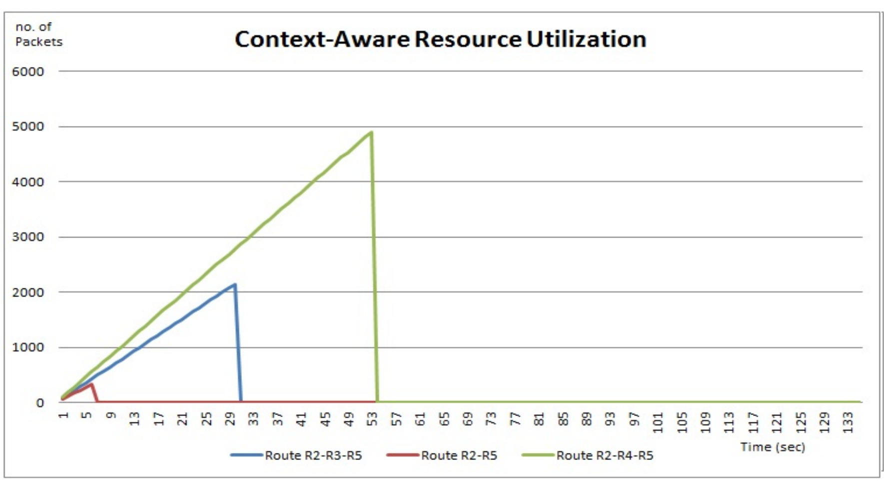

Figure 18 Context-Aware Utilization

The shortest path method, on the other hand, used only a direct path and transferred files one after another, taking more time as shown in Figure 19. This resulted in idle links and idle traffic, resulting in a transfer time of around 133 seconds.

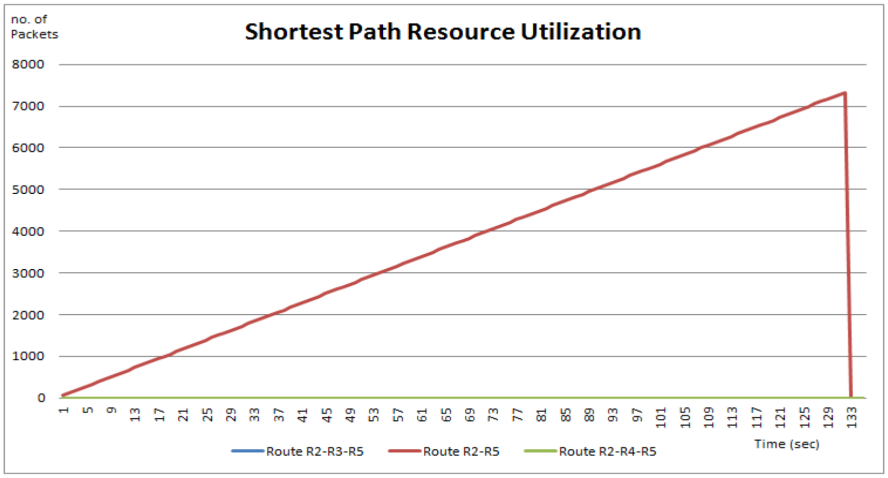

Figure 19 Shortest Path Utilization

The analysis favored the context-aware algorithm as it intelligently used resources and made full use of available bandwidth. The shortest path method, on the other hand, ignored high bandwidth links and prioritized the shortest path, which proved to be costly in terms of delays.

5. Conclusion and Future work

The research suggests that the shortest path method of routing is effective when there are no packet varieties or diverse networks. However, when there are various resources, a context-aware algorithm can be more robust and efficient. This method can avoid unwanted delays and collisions, which can significantly impact transmission. Additionally, using a context-aware algorithm allows for stateless routing, allowing for Quality-of-Service implementation without maintaining the per flow state of routers. This dynamic forwarding mechanism provides more scalability and can be used to identify low-cost and delay-constrained paths. The analysis of different routing algorithms demonstrated that context-aware algorithms were superior in terms of cost-effectiveness, delays, and throughput. In complex networks, routing decisions are influenced by various characteristics, making context-aware algorithms a more suitable choice. Overall, the research highlights the importance of considering various factors when choosing a routing method.

This research highlights the need for further development in the context-aware routing framework. The current method is not fully functional and does not include collision count in statistics. Future work could involve transferring files reality, calculating collisions and retransmissions caused in both networks. The algorithm is acceptable for static networks but needs more work to adapt to network changes and respond to shutdowns or increased bandwidth. Answers to these questions will be sought in future years and should focus on improving the algorithm's performance in dynamic networks.

Referrences

[1] ter Mors, A., Zutt, J., & Witteveen, C. (2023, September). Context-Aware Logistic Routing and Scheduling. In ICAPS (pp. 328-335).

[2] ter Mors, A., van Belle, J., & Witteveen, C. (2023, May). Context-aware multi-stage routing. In Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1 (pp. 49-56).

[3] Ahn, C. W., & Ramakrishna, R. S. (2002). A genetic algorithm for shortest path routing problem and the sizing of populations. IEEE transactions on evolutionary computation, 6(6), 566-579.

[4] Masillamani, M. R., Suriyakumar, A. V., Ponnurangam, R., & Uma, G. V. (2000). Genetic algorithm for distance vector routing technique. AIML, 6, 160-163.

[5] Lammle, T. (2016). CCENT ICND1 Study Guide: Exam 100-105. John Wiley & Sons.

[6] Mansfield, K. C., & Antonakos, J. L. (2009). Computer networking for LANS to WANS: hardware, software and security. Delmar Learning.

[7] Valarmathi, A., & Chandrasekaran, R. M. (2010). Congestion aware and adaptive dynamic source routing algorithm with load-balancing in MANETs. International Journal of Computer Applications, 8(5), 1-4.

[8] ter Mors, A. (2007). Robustness in Context-Aware Route Planning.

[9] GNH Irani, H Izadkhah - Computer Standards & Interfaces, 2024 - Elsevier

[10] A Das, A Leaf, CA Varela… - … on cloud computing …, 2020 - ieeexplore.ieee.org

[11] https://rmkcet.ac.in/docs/regulation/ADS%20R2022.pdf

[12] XL Li, KQ Chen, HJ Huang, S Yang… - Propellants …, 2024 - Wiley Online Library

[13] GKL Rao, N Mokhtar - … of Research on Instructional Technologies in …, 2023 - igi-global.com

[14] OB Yahia, Z Garroussi, O Bélanger, B Sansò… - arXiv preprint arXiv …, 2023 - arxiv.org

[15] MM Karbassian, H Ghafouri-Shiraz - 2023 - Wiley Online Library