SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Design And Performance Evaluation Of An Adaptive Geographic Routing Framework For Reliable And Efficient Communication In Multi-Radio Mobile Ad Hoc Networks

Dr. Anand M[1], Girijamba D L[2], *Dr. Deepa S N[3], Dr. Likhitha R[4], Dr. Vishwesh J[5], Goutham V[6]

[1]Associate Professor, Department of Information Science and Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Karnataka, India - 570016, anandm@gsss.edu.in
[2] Assistant Professor, Department of Electronics and Communication Engineering Vidyavardhaka College of Engineering, Mysuru, Karnataka, India- 570002, girijamba@vvce.ac.in
[3]Assistant Professor, Department of Electrical and Electronics Engineering, SJCE, JSSSTU, Mysuru, Karnataka, India- 570006, deepa.archak@jssstuniv.in
[4] Assistant Professor, Department of Electrical and Electronics Engineering, Nitte(Deemed to be University), Nitte Meenakshi Institute of Technology (NMIT), Bengaluru, Karnataka, India- 560064, likhitha.r@nmit.ac.in
[5] Associate Professor, Department of Computer Science & Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Karnataka, India 570016, vishweshj@gsss.edu.in
[6] Assistant Professor, Department of Electronics and Communication Engineering, BGS Institute of Technology, Adichunchangiri University, Mandya, Karnataka, India-571448, gouthamv77@gmail.com
Corresponding Author Mail id: deepa.archak@jssstuniv.in

Abstract: Mobile Ad Hoc Networks (MANETs) with multiple radio interfaces are emerging as a promising communication technology for military and battlefield applications, where reliable and efficient data transmission is essential. This paper presents a novel geographical routing protocol for heterogeneous multi-radio, multi-band tactical MANETs. The proposed protocol utilizes three radio interfaces, namely IF0, IF1, and IF2, with each interface operating on a different frequency band. Multiple routing metrics are considered to dynamically select the most suitable interface and next-hop node according to network conditions and traffic requirements. The geographical information of neighboring nodes is also utilized to improve packet forwarding toward the destination. The proposed protocol is implemented and evaluated using the JProwler simulation platform, with Java Swing programming developed using NetBeans IDE on the Windows operating system. The simulation results demonstrate that the proposed approach improves communication reliability, routing efficiency, and Quality of Service (QoS) compared with conventional GPRS-based communication. The proposed protocol therefore provides an effective approach for reliable and adaptive communication in dynamic heterogeneous multi-radio tactical MANET environments.

Keywords: Tactical ad hoc networks, MANETS, Geographical routing

1. INTRODUCTION

A novel multi-metric geographical routing protocol is proposed for next-generation Tactical Multi-hop Mobile Radio (TMMR) networks operating in heterogeneous multi-radio environments. Unlike conventional geographical routing protocols that typically rely on a single radio interface, the proposed approach considers nodes equipped with multiple radio interfaces, each operating over a distinct frequency band. Consequently, the transmission range and communication characteristics may vary across interfaces [1]. To maintain updated neighborhood information, nodes periodically exchange Hello messages containing their geographical information, enabling efficient neighbor discovery without introducing significant additional control overhead [2]. Conventional geographical routing protocols face several challenges in multi-radio environments, particularly when different interfaces provide heterogeneous communication ranges and network characteristics. Most existing location-based routing schemes are primarily designed for single-radio networks and therefore have limited capability to efficiently exploit multiple radio interfaces. Tactical Information Communication Networks (TICN), which

constitute a critical component of the Network-Centric Warfare (NCW) infrastructure, require robust, reliable, and efficient communication mechanisms to support mission-critical operations [5].
Military communication networks impose more stringent requirements on packet delivery reliability, end-to-end delay, and Quality of Service (QoS) than conventional commercial networks. Therefore, an effective routing mechanism must identify suitable communication paths while satisfying the QoS requirements of diverse tactical applications. Furthermore, next-generation tactical wireless networks need to operate effectively in highly dynamic environments by intelligently selecting the most appropriate radio interface according to the requirements of the application and the characteristics of the transmitted data [3].To address these challenges, the proposed routing mechanism integrates multiple routing metrics and dynamically selects the most suitable radio interface based on network conditions and application-specific requirements [4]. By combining geographical routing with efficient multi-interface selection, the proposed scheme aims to enhance communication reliability, transmission efficiency, and overall QoS performance in next-generation TMMR networks.

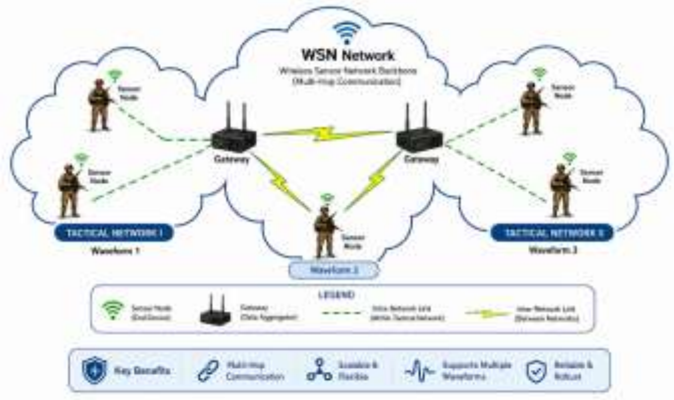

Figure 1: TMMR networking structure in tactical information communication network

2. LITERATURE REVIEW

Wireless Sensor Networks (WSNs) and Mobile Ad Hoc Networks (MANETs) are widely used in applications such as environmental monitoring, healthcare, industrial automation, military communication, and Internet of Things (IoT) systems. Routing is a critical component of these networks because the nodes generally operate with limited energy, processing capability, bandwidth, and storage resources. Khan et al. [1] presented a comprehensive survey of energy-efficient routing protocols for WSNs and highlighted the importance of reducing communication overhead and balancing energy consumption among sensor nodes. Their study showed that efficient routing can significantly extend network lifetime, although maintaining energy efficiency becomes challenging when network topology and traffic conditions change dynamically.
Quality of Service (QoS) has also become an important consideration in wireless routing. Kumar and Tripathi [2] investigated machine learning-based QoS-aware routing in MANETs and demonstrated that learning techniques can be used to improve routing decisions according to parameters such as delay, bandwidth, and network conditions. Liu et al. [3] focused on secure and efficient routing in WSNs, emphasizing that routing protocols must address security threats while maintaining acceptable energy consumption and network performance. Their work indicates that security mechanisms can introduce additional communication and computational overhead, creating a trade-off between network protection and resource efficiency.
Geographic and location-based routing techniques have been explored to improve routing efficiency in wireless ad hoc networks. Alshamrani et al. [4] investigated optimized geographic routing protocols and emphasized the use of node-location information to select efficient forwarding paths. Such approaches can reduce unnecessary routing information and improve scalability. However, their effectiveness may be affected by inaccurate location information, mobility, and unreliable wireless links. Gupta and Kumar [5] examined machine learning-based routing protocols for WSNs and demonstrated the potential of intelligent techniques to adapt routing decisions according to changing network conditions. The study indicates that machine learning can provide better adaptability than traditional static routing approaches, although computational complexity and training requirements remain important concerns.
Load balancing is another significant factor in maintaining network performance. Elazhary [6] studied improved load-balancing techniques for MANETs and highlighted the need to distribute traffic effectively among available nodes. Uneven traffic distribution can cause particular nodes to become congested or deplete their energy rapidly, resulting in reduced network lifetime and connectivity. Chen et al. [7] investigated AI-driven routing optimization for next-generation wireless networks and discussed how

artificial intelligence can support dynamic routing decisions. Their work reflects the growing shift from conventional rule-based routing toward intelligent approaches capable of adapting to variations in traffic, topology, and network requirements.

Security and trust management have increasingly been incorporated into routing protocols. Sharma and Kim [8] explored blockchain-based secure routing in MANETs, demonstrating the potential of decentralized mechanisms for improving trust and security during routing. Although blockchain can provide stronger integrity and trust management, its computational, storage, and communication requirements may be unsuitable for highly resource-constrained sensor nodes. Verma and Sangaiah [9] investigated energy-aware clustering-based routing for IoT-oriented WSNs and emphasized the importance of efficient cluster formation and communication management. Clustering can reduce long-distance transmissions and improve energy utilization, but selecting suitable cluster heads and maintaining clusters under changing network conditions remain challenging.

Recent research has increasingly focused on deep learning and reinforcement learning for autonomous routing. Wang et al. [10] studied deep reinforcement learning-based routing for wireless networks, demonstrating how learning agents can identify routing decisions based on network states and performance objectives. Such methods can potentially optimize multiple parameters simultaneously, including throughput, delay, congestion, and energy consumption. However, deep reinforcement learning generally requires substantial training data and computational resources, which may limit direct deployment on constrained wireless nodes. Nguyen and Kim [11] examined adaptive QoS routing in tactical ad hoc networks and highlighted the importance of dynamically responding to changing network conditions and application requirements. Adaptive routing is particularly important in environments where topology, traffic load, and link quality can vary rapidly.

With the emergence of 6G networks, routing requirements are becoming more complex. Zhang et al. [12] discussed next-generation routing protocols for 6G-enabled wireless networks and emphasized the need for highly adaptive, intelligent, and scalable routing mechanisms. Future wireless networks are expected to support heterogeneous devices, massive connectivity, low-latency applications, and dynamic communication environments. Patel and Singh [13] proposed a hybrid routing approach for delay-sensitive MANET applications, demonstrating the importance of combining routing strategies to satisfy application-specific requirements. Hybrid protocols can exploit the advantages of different routing mechanisms, although they may require more sophisticated decision-making and protocol management.

More recent research has concentrated on simultaneously optimizing energy consumption and QoS through artificial intelligence. Malik et al. [14] investigated AI-based energy and QoS optimization in WSNs and highlighted the potential of intelligent algorithms to balance multiple network objectives. This direction is particularly relevant because optimizing only one parameter, such as energy consumption, may negatively affect other performance indicators such as delay, packet delivery ratio, or throughput. AI-based multi-objective optimization therefore provides a promising direction for developing adaptive routing protocols capable of responding to dynamic network conditions.

Overall, the literature demonstrates a clear evolution from conventional energy-aware and geographic routing protocols toward adaptive, machine-learning-based, AI-driven, and security-aware routing mechanisms. Existing studies have addressed important aspects such as energy efficiency [1,5,9], QoS optimization [2,11,14], security [3,8], load balancing [6], geographic routing [4], and intelligent routing optimization [7,10,12]. However, several challenges remain, including communication overhead, computational complexity, packet loss, unreliable links, dynamic topology changes, scalability, and limited node resources. In particular, many existing approaches optimize routing selection but provide limited attention to efficiently recovering or identifying complete packet paths when routing information is incomplete or packets are lost. This indicates a research gap for developing lightweight and robust path-information techniques that can operate effectively in dynamic WSN environments while maintaining low communication overhead, high accuracy, energy efficiency, and resilience to topology changes.

3. SYSTEM DESIGN

Design is a crucial phase in software development, as an effective design forms the foundation of a successful system. System design can be defined as the process of applying appropriate techniques and principles to define a system in sufficient detail so that it can be implemented physically. The design specification explains the overall features of the system, its components, and the way these components interact with end users.

3.1 Fundamental Design Concepts

Over the years, several fundamental design concepts have evolved and become essential in software engineering. These concepts provide a strong framework for developing reliable and efficient systems. Concepts such as abstraction, refinement, modularity, software architecture, control hierarchy, structural partitioning, data structures, software procedures, and information hiding have been incorporated into this project to ensure that the system satisfies the required specifications effectively.

3.1.1 Input Design

Input design refers to the process of transforming user-oriented data into a computer-readable format. The primary objective of input design is to make data entry simple, accurate, and efficient while minimizing errors. To achieve this, the system incorporates user-friendly interfaces, consistent data formats, and interactive features that assist users during data entry.

Special attention is given to input design because data collection is often one of the most critical and expensive stages in system development. In this project, users can send data packets to the destination machine using known IP addresses. If the IP address is unknown or entered incorrectly, errors may occur during transmission. Therefore, proper validation and guidance mechanisms are included to ensure accurate input handling.

3.1.2 Output Design

Output design focuses on presenting processed information clearly and effectively to end users. A good output should satisfy user requirements while ensuring clarity and reliability. Outputs act as the primary communication channel between the system and the user.

In this system, output design is particularly important because it ensures that the data packets received by the destination are identical to those sent by the source, without corruption or spoofing. The generated outputs also help maintain permanent records for future reference. Efficient output handling improves communication reliability between source and destination machines and enhances the overall system performance.

3.1.3 MVC Design Method

The system follows the Model-View-Controller (MVC) design methodology. In Swing applications, a simplified variation of MVC known as the model-delegate architecture is commonly used. In this approach, the view and controller functionalities are combined into a single component called the UI delegate.

Each Swing component consists of two major parts: the model and the UI delegate. The model is responsible for maintaining the state and data of the component, while the UI delegate manages the visual representation and handles user interactions and GUI events. The UI delegate works together with the Abstract Window Toolkit (AWT) to process events and update the component accordingly.

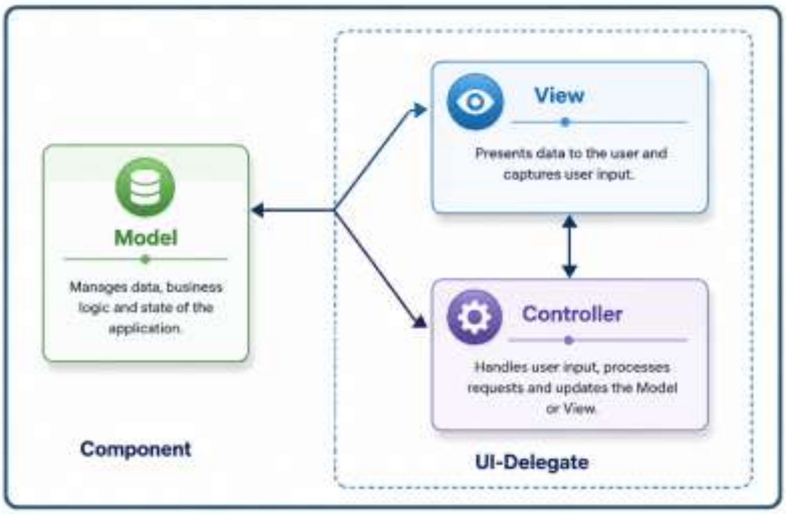

Fig 2 Combination of View & Controller into a UI delegate object

The design method that has been followed to design the architecture of the system is MVC design pattern. Swing uses the model-view-controller (MVC) architecture as the fundamental design behind each of its components. Essentially, MVC breaks GUI component into three elements. Each of these elements plays a crucial role in how the component behaves. The MVC design pattern separates a software component into three distinct pieces: a model, a view, and a controller.

4. SYSTEM ANALYSIS

System analysis is the process of identifying problems, understanding system requirements, and determining the most suitable solution. It involves studying the existing system, defining objectives, evaluating possible solutions, and applying appropriate technologies to solve the identified issues.

Feasibility study plays a significant role in system analysis as it helps determine whether the proposed system can be successfully designed and implemented.

4.1 Feasibility Study

A system can theoretically be considered feasible if unlimited time and resources are available. However, in practical scenarios, resource limitations and time constraints make feasibility analysis essential. Conducting feasibility studies at an early stage helps avoid unnecessary expenditure, wasted effort, and implementation failures. Feasibility and risk analysis are closely related because higher project risks directly affect the successful development of quality software.

The feasibility study for this project focuses mainly on performance and technical feasibility.

4.1.1 Performance Analysis

For the successful execution of the project, a reliable networking environment is required. The system is implemented on the Windows XP operating system. The primary objective of this project is to reduce the impact of black hole attacks in the network by routing packets through random paths.

Performance analysis is carried out to evaluate whether malicious nodes are still capable of intercepting packets. To measure the effectiveness of the proposed approach, a parameter known as Packet Interception Probability (PIP) is used. The analysis helps determine the reliability and security of packet transmission within the network environment.

4.1.2 Technical Analysis

A system becomes beneficial only when it can satisfy the technical requirements of the organization. Technical feasibility determines whether the proposed system can be successfully developed, implemented, and operated without major technical difficulties.

The following aspects are considered during technical analysis:

- **System Changes:**
  The proposed system introduces positive improvements such as increased efficiency and enhanced service quality.
- **Required Skills:**
  The tools and technologies used in this project are widely adopted in the software industry, making skilled professionals readily available.
- **Acceptability:**
  The system architecture is designed to be user-friendly and flexible, ensuring ease of use and smooth adoption by users.

5. Implementation

Implementation is the phase in which the theoretical design is transformed into a functional system. During this stage, the responsibility shifts from system development to practical usage by the end users. Proper planning and effective management are essential for successful implementation, as a poorly executed implementation process may result in confusion, errors, and operational difficulties. The implementation phase involves careful project planning, investigation of system requirements and constraints, designing suitable methods for system changeover, evaluating different changeover techniques, selecting an appropriate platform, and choosing a suitable programming language for system development.

5.1 Language Used for Implementation

The implementation phase requires the system design to be accurately converted into executable code using an appropriate programming language. Selecting a suitable programming language is important because an unsuitable language may affect the efficiency, reliability, maintainability, and overall performance of the system. In this project, Java is selected as the programming language because of its platform independence and strong support for object-oriented programming. Java also provides a rich standard library that simplifies application development and offers familiar C++-like syntax, making it relatively easy to learn and use. Other important features of Java include automatic garbage collection, which manages memory efficiently, and Swing support, which provides components for developing graphical user interfaces. Java also provides applet support, which was traditionally used for developing small interactive applications that could run within web browsers.

5.2 Platform Used for Implementation

A platform provides the environment required for developing and executing software applications. In this project, the Windows XP operating system is used as the implementation platform because of its stability, networking capabilities, security features, and user-friendly environment. Windows XP provides integrated networking support that enables systems to communicate efficiently over networks. It also offers improved stability and security features, along with remote desktop connectivity for accessing systems remotely. The operating system provides enhanced device driver verification and reduces the need for frequent system reboots. It also includes improved code protection, side-by-side DLL support, and Windows File Protection to help maintain system integrity.

Windows XP uses a preemptive multitasking architecture, allowing multiple applications and processes to execute efficiently. It provides scalable memory and processor support and includes security features such as the Encrypting File System (EFS), IP Security (IPSec), and Kerberos authentication. In addition, support for smart cards provides enhanced authentication capabilities. Windows Firewall and Security Center further improve system security by helping protect the system from unauthorized network access and security threats. The user-friendly graphical interface also makes the platform convenient for users and developers during system development, testing, and execution.

6. TESTING

Testing is the systematic process of verifying and validating the developed software to ensure that it performs according to the specified requirements. It confirms that the individual components of the system function correctly and that all modules work together as intended. Testing is an essential stage of software development and serves as the final verification process before the system is deployed for practical use. The main objectives of testing are to ensure software quality, identify and eliminate remaining errors, validate the system against the original requirements, and ensure its reliability, stability, and correctness. During the testing process, particular attention is given to examining the source code, identifying defects, and refining the system to achieve the expected performance.

6.1 Unit Testing

Unit testing is the process of testing individual modules or components of a software system independently to verify their correctness. It is also referred to as module testing. Each module is tested separately to ensure that it performs its intended function accurately before it is combined with other modules. Unit testing primarily focuses on verifying the functionality of individual modules, examining internal control paths, detecting coding and logical errors at an early stage, and ensuring that the expected outputs are produced for the given inputs. By validating each functional component independently, unit testing simplifies the subsequent stages of testing and helps reduce errors during system integration.

6.2 Integration of System Components

After the successful completion of unit testing, the individual modules and functions are integrated to form the complete system. The integration process is performed systematically to ensure that the components communicate correctly and work together without affecting their individual functionalities.

Integration of Functions into Classes: Initially, individual functions are developed and tested independently. Once their correctness has been verified, related functions are grouped and integrated into their respective classes. This ensures that the functions operate correctly within the class structure and that the required relationships between them are maintained.

Integration of Different Classes: After individual classes have been developed and tested, they are combined to form the overall application structure. The interaction between the classes is then tested to ensure that data is transferred correctly and that the classes perform their respective operations without conflicts or errors.

Integration of Front-End with Back-End: The graphical user interface of the project is developed using Java Swing. The backend logic is then integrated with the front-end to provide the required user interaction and system functionality. The integrated application is tested to verify proper communication between the user interface and the underlying system logic and to ensure that all operations are executed correctly.

6.3 Integration Testing

Integration testing is performed to verify that the different modules and components of the system work together correctly after integration. It focuses on identifying errors that may occur due to interactions or interfaces between individual modules. This testing ensures proper communication among components, maintains data integrity, and verifies that combined functionalities produce the expected results. The major objectives of integration testing are to detect interface-related errors, verify communication and interaction between modules, ensure data consistency across different components, and validate the overall functionality of the integrated system. Once all the modules have been successfully integrated and tested, higher-level testing is performed on the complete application.

Top-Down Integration: Top-down integration testing begins with the main or highest-level program module and progressively incorporates the lower-level modules. The integration process continues systematically until all modules are connected and tested. The testing sequence can follow either a depth-first or breadth-first approach, depending on the structure of the application. This method helps verify the overall control flow of the system from the main module toward the supporting modules.

Bottom-Up Integration: Bottom-up integration testing starts with the lowest-level modules and gradually integrates them with higher-level modules. The testing process proceeds upward until the complete system is assembled. Since the lower-level modules are tested and made available first, this approach reduces or eliminates the need for temporary stubs. It is particularly useful for verifying the correctness of lower-level processing before integrating it into the higher-level system components.

7. RESULTS

The simulation results demonstrate the effectiveness of the proposed system.

7.1 End-to-End Delay

The analysis results show that the end-to-end delay in the proposed multi-metric routing approach is significantly lower when compared with existing GPRS technology. Reduced delay indicates improved communication efficiency and better overall network performance. Therefore, the proposed multi-metric technique provides superior performance compared to traditional GPRS-based systems.

Fig 3: Illustration of the output End to End delay

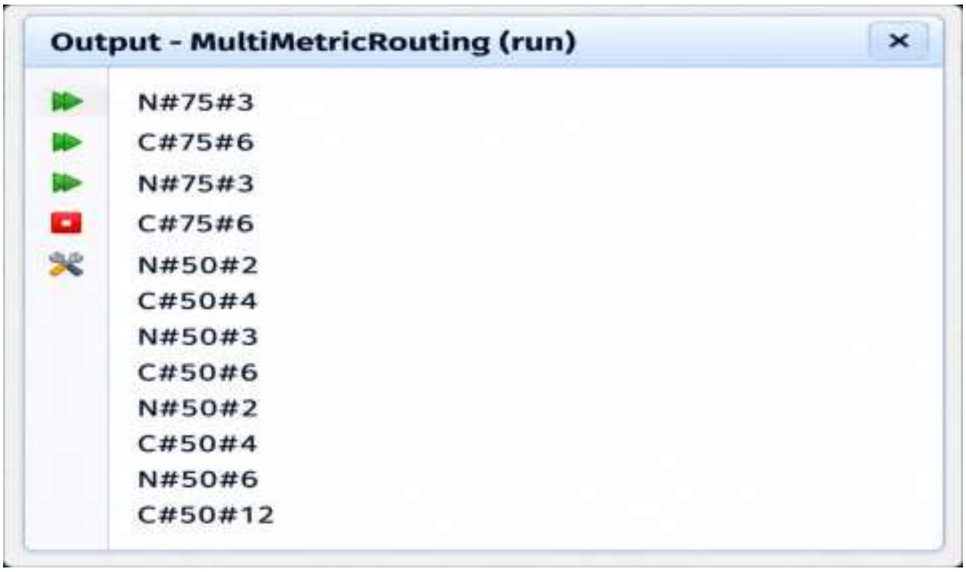

Figure 4: Output - multi metric routing of End to End delay

7.2 Packet Delivery Ratio :

As shown in the figure 8.3 and figure 8.4, the Ratio of the packet delivery is very high in multi metric while comparing to the GPRS technology. Hence the performace of this multimetric technology is very high comparing to the existing GPRS technology.

Figure 5: Illustration of the output Packet Delivery Ratio

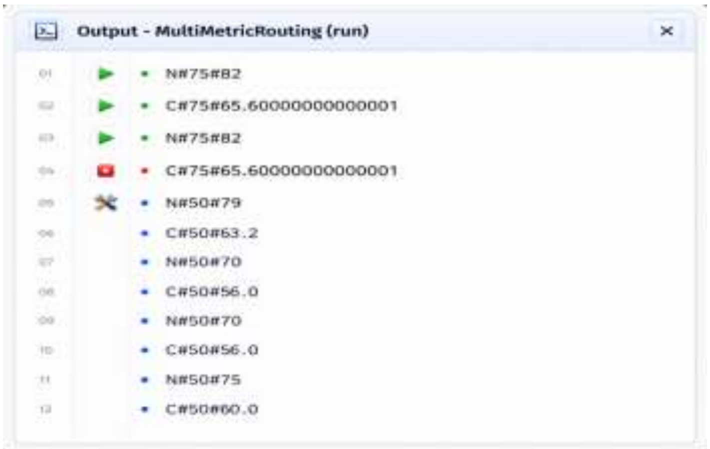

Figure 6: Output - multi metric routing of Packet Delivery Ratio

7.3 Throughput :

As shown in the figure 8.5 and figure 8.6, the throughput is very high in multi metric while comparing to the GPRS technology. Hence the performace of this multimetric technology is very high comparing to the existing GPRS technology.

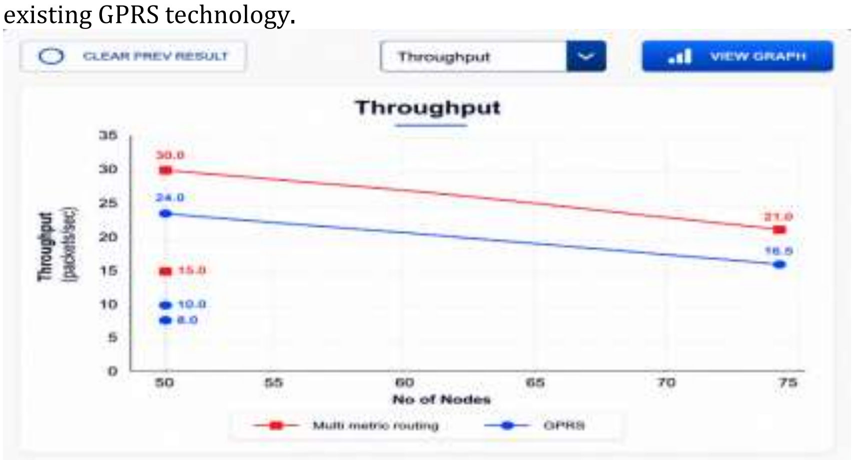

Figure 7: Illustration of the Throughput

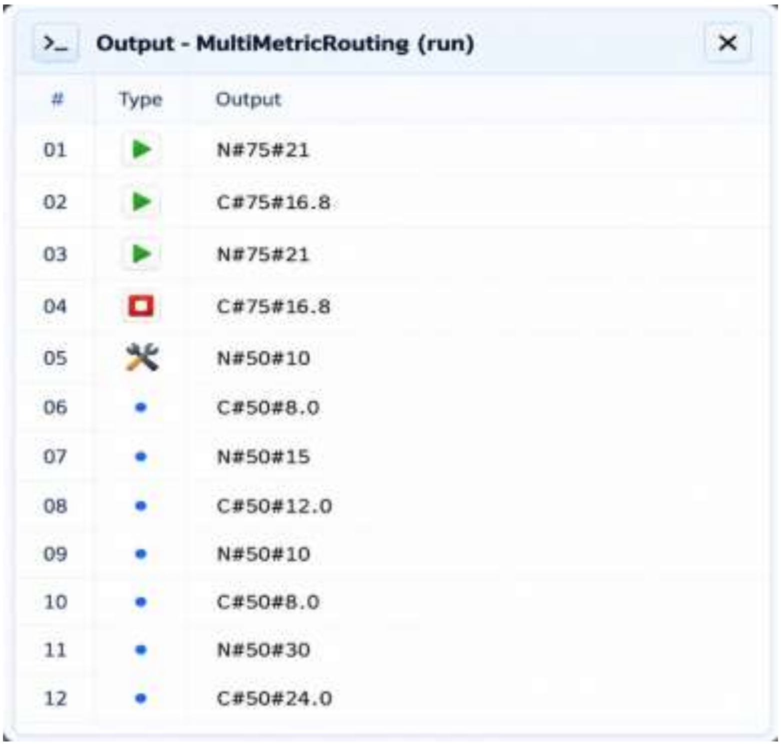

Figure 8: Output - multi metric routing of Throughput

CONCLUSION & FUTURE ENHANCEMENT

Conclusion

A new multi-metric geographical routing protocol has been proposed for tactical ad hoc networks to address the diverse communication requirements of tactical environments. The proposed protocol utilizes multiple routing metrics to efficiently handle different types of traffic, including voice, video, and short message services. Based on the calculated routing metrics, the protocol dynamically selects the most appropriate interface from the available network interfaces, thereby improving the efficiency and reliability of data transmission.The proposed approach uses geographical information available at the source node, including the location and distance of neighboring nodes, to determine an appropriate forwarding node. Data packets are forwarded through the neighboring node that is geographically closest to the destination. Since the protocol follows a geographical routing strategy, it does not require additional route discovery control packets such as Route Request (RREQ) and Route Reply (RREP) messages. Eliminating these control packets significantly reduces routing overhead and helps improve the overall efficiency of the network.

The performance of the proposed multi-metric routing protocol was evaluated through simulation and compared with the conventional Greedy Perimeter Stateless Routing (GPSR) protocol. The simulation results demonstrate that the proposed approach achieves better performance in terms of key network parameters, including end-to-end delay, packet delivery ratio, and throughput, while maintaining low routing overhead.The results show that the proposed multi-metric routing approach significantly reduces end-to-end delay compared with GPSR, enabling faster delivery of packets across the network. In addition, the proposed protocol achieves a higher packet delivery ratio, indicating improved reliability and successful transmission of data packets. The throughput is also considerably higher, demonstrating more efficient utilization of available network resources. Overall, the simulation results confirm that the proposed multi-metric geographical routing protocol provides improved performance, reliability, and efficiency for tactical ad hoc communication networks compared with the traditional GPSR protocol.

FUTURE ENHANCEMENT

The proposed routing protocol can be further enhanced by incorporating machine learning and artificial intelligence techniques to dynamically select the most suitable routing metric based on changing network conditions. Future work can focus on developing adaptive mechanisms for real-time traffic classification, congestion prediction, link-quality estimation, and energy-aware routing. The protocol can also be extended to support larger and highly dynamic tactical ad hoc networks, multiple simultaneous interfaces, and heterogeneous communication technologies. Security mechanisms such as intrusion detection and trust-aware routing can be integrated to improve resilience against malicious nodes and attacks. Further validation through real-world testbeds and large-scale simulations can help evaluate scalability, reliability, latency, throughput, and energy efficiency under realistic tactical communication scenarios.

REFERENCES

[1] M. A. Khan, A. Rehman, and T. A. Zia, "Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey," IEEE Access, vol. 8, pp. 107527–107542, 2020.

[2] S. Kumar and R. Tripathi, "QoS-Aware Routing in MANETs Using Machine Learning Techniques," IEEE Internet of Things Journal, vol. 8, no. 3, pp. 1527–1538, 2021.

[3] Y. Liu, J. Li, and M. Guizani, "Secure and Efficient Routing Protocols for Wireless Sensor Networks," IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 221–245, 2021.

[4] A. Alshamrani et al., "Optimized Geographic Routing Protocols for Wireless Ad Hoc Networks," IEEE Access, vol. 9, pp. 32045–32060, 2021.

[5] R. Gupta and N. Kumar, "Machine Learning-Based Routing Protocols for Wireless Sensor Networks," Future Generation Computer Systems, vol. 124, pp. 414–427, 2022.

[6] H. Elazhary, "Improved Load Balancing Techniques in Mobile Ad Hoc Networks," Wireless Networks, vol. 28, pp. 1123–1138, 2022.

[7] M. Chen, Y. Miao, and X. Wang, "AI-Driven Routing Optimization in Next-Generation Wireless Networks,"
IEEE Network, vol. 36, no. 2, pp. 120–126, 2022.

[8] P. K. Sharma and D. I. Kim, "Blockchain-Based Secure Routing in MANETs," IEEE Transactions on Network Science and Engineering, vol. 10, no. 1, pp. 89–102, 2023.

[9] S. Verma and A. K. Sangaiah, "Energy-Aware Clustering Routing Protocol for IoT-Based WSN," Sustainable Computing: Informatics and Systems, vol. 36, 2023.

[10] J. Wang et al., "Deep Reinforcement Learning-Based Routing for Wireless Networks," IEEE Transactions on Wireless Communications, vol. 22, no. 5, pp. 3456–3469, 2023.

[11] T. Nguyen and H. Kim, "Adaptive QoS Routing in Tactical Ad Hoc Networks," IEEE Access, vol. 11, pp. 55432–55445, 2023.

[12] L. Zhang, Q. Wu, and X. Chen, "Next-Generation Routing Protocols for 6G-Enabled Wireless Networks," IEEE Communications Magazine, vol. 61, no. 1, pp. 64–70, 2024.

[13] A. Patel and S. Singh, "Hybrid Routing Protocol for Delay-Sensitive Applications in MANETs," Computer Networks, vol. 235, 2024.

[14] K. R. Malik et al., "Energy and QoS Optimization in Wireless Sensor Networks Using AI Techniques," IEEE Access, vol. 12, 2024.

[15] Dr. Anand M et al., "A Novel Energy Efficient Approach for Reliable Path Recovery and Fault Tolerant Communication in Dynamic Wireless Sensor Networks", International Journal of Computer Information Systems and Industrial Management Applications, Volume18, Issue 4s, July 2026, pp. 237-244