



Article

An Improved Routing Protocol for Optimum Quality of Service in Device-to-Device and Energy Efficiency in 5G/B5G

Sanusi Mohammad Bunu , Omar Younis Alani * and Mohammad Sarae

School of Science, Engineering and Environment, University of Salford, Manchester M5 4WT, UK; s.m.bunu@edu.salford.ac.uk (S.M.B.); m.saraee@salford.ac.uk (M.S.)

* Correspondence: o.y.k.alani@salford.ac.uk

Abstract: Some challenges when implementing the optimized link state routing (OLSR) protocol on real-life devices and simulators are unmanageable: link quality, rapid energy depletion, and high processor loads. The causes of these challenges are link state processing, unsuitable multipoint relay (MPR) nodes, and information base maintenance. This paper proposes a structured, energy-efficient link sensing and database maintenance technique. The improved OLSR in the paper replaces the OLSRv2's HELLO, HELLO, and Topology Control (TC) message sequence with a new sequence. MPR nodes are not mandated to broadcast TC messages if the number of nodes and their OLSRv2 addresses remain unchanged after subsequent broadcasts or if no node reported 2-hop symmetric connections. The paper further proposes an MPR selection technique that considers four parameters: node battery level, mobility speed, node degree, and connection to the base station for optimum relay selection. It combines the four parameters into one metric to reduce energy dissipation and control routing overhead. The modifications were implemented in NS-3, and the simulation results show that our improved OLSR outperforms the existing OLSR, OLSRv2 and other improved routing protocols in energy consumption, routing overhead, the packet delivery ratio and end-to-end delay, as compared to the related literature.

Keywords: algorithm; device-to-device; energy; quality of service; OLSR



Citation: Bunu, S.M.; Alani, O.Y.; Sarae, M. An Improved Routing Protocol for Optimum Quality of Service in Device-to-Device and Energy Efficiency in 5G/B5G. *Future Internet* **2024**, *16*, 347. <https://doi.org/10.3390/fi16090347>

Academic Editors: Jose I. Moreno Novella and Gianluigi Ferrari

Received: 24 July 2024

Revised: 3 September 2024

Accepted: 10 September 2024

Published: 23 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

D2D communication systems are one of the fundamental elements of 5G and B5G mobile telecommunication networks. Unlike traditional cellular networks, with D2D communication technology, nodes can communicate with one another with or without the involvement of a base station [1]. Direct communication between devices nearby will achieve low latency and high throughput compared to the communication of devices through the nearest eNodeB, which can be congested due to high traffic load. The potential of D2D communication to transform next-generation cellular networks has resulted in the integration of the technology in many areas, including networks for public safety [2], proximity-based services [3], vehicular networks [4], multi-hop relaying [5], and cellular offloading [6]. Given the vigorous nature of D2D communications, routing protocols are an architecture necessary for implementing efficient D2D communication systems [7].

Therefore, the network's overall performance will be negatively affected if a wrong and ineffective routing protocol is used for D2D communication systems. Routing protocols serve as a critical architecture required for creating an effective D2D communication system due to the robust nature of D2D communication [7]. As a result, researchers have used different techniques to optimize the traditional routing protocols, focusing mainly on energy efficiency, scalability, reliability, and enhanced resource utilization [8–10]. A specific problem with D2D communication is the issue of wireless device relays and exhaustible energy sources. Therefore, energy efficiency and less overhead routing protocols are necessary to improve network performance. However, in outbound autonomous D2D

communication, there is no central administrative body, and nodes control their mobility. This can lead to frequent changes in selected multipoint relay (MPR) and communication links, challenging traditional routing techniques [7,11].

Conventionally, routing protocols for D2D/wireless devices are categorized as proactive, reactive and hybrid, depending on network structure, routing strategy, communication mode, and state of information [12–14]. Proactive or table-driven routing protocols store routing information in tables and require nodes to establish communication links in advance [15]. This allows nodes to maintain an updated network topology, making finding available routes to desired destinations easier. Proactive routing protocols like OLSR [15] and DSDV [16] are suitable for low-latency applications due to their quick connectivity and minimum delay. Reactive routing protocols are also known as on-demand routing techniques. Reactive routing techniques require nodes to create a path to a destination node only when needed [17,18]. The most famous and widely used reactive protocols are the Ad Hoc On-Demand Distance Vector (AODV) [19] and Dynamic Source Routing (DSR) [20]. The general concept around hybrid routing techniques is to combine the idea of both reactive and proactive routing protocols with an effort to overcome some weaknesses. The most famous and widely used hybrid routing protocols are the Zone Routing Protocol (ZRP) [21] and hierarchical routing techniques [22]. The proactive routing protocol in OLSRv2 raises several difficulties, including those related to energy consumption, routing overhead, and the best choice of relays [23–25]. This is because link state processing, information base maintenance, and low battery energy nodes selected as multipoint relays experience a rapid power failure and create network partitions. Despite these challenges, proactive routing protocols remain vital for future communication technologies prioritizing low latency [26], which this paper thoroughly considers for OLSR modification.

As stated earlier, link sensing, MPR nodes, and information base maintenance are both the strengths and Achilles’ heels of OLSRv2 [24,27–29]. The existing OLSRv2 specification assumes mobile devices have infinite memory, processing abilities, and ample storage. Unfortunately, mobile devices are not base stations that use redundant processing, storage, and backup generators. Figure 1 below succinctly captures a typical OLSRv2 mobile scenario comprising four mobile nodes gradually drifting apart. Broadcasts from node A are heard by node B. However, node B’s broadcasts are unable to reach node A. Thus, the link between nodes A and B is asymmetric. Nodes B and D are two-hop neighbors because node C has a symmetric link with both. Node C is equally a one-hop neighbor of both nodes B and D. Although the nodes are drifting apart, OLSRv2 expects each node to broadcast HELLO messages at a HELLO_INTERVAL(s) of 2 s, which will continually trigger table updates and dissipate energy, even when nodes are outside each other’s broadcast range.

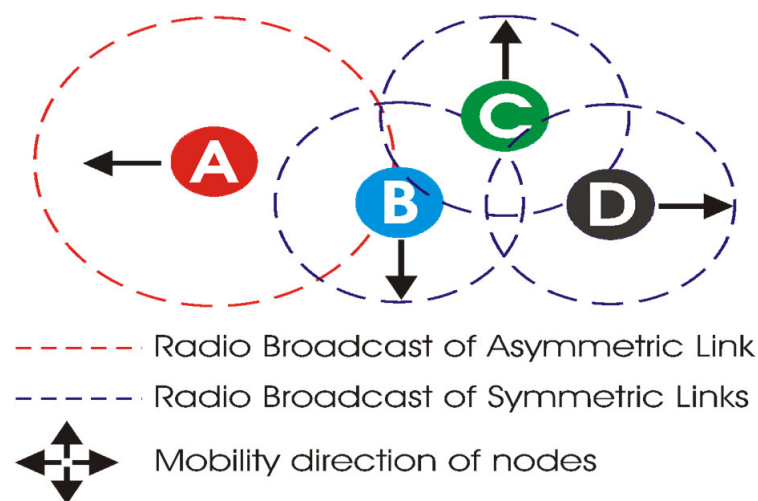


Figure 1. Group of OLSRv2 nodes (Letters A-D represent examples of nodes).

A careful analysis of the opposing arguments in [30] revealed that OLSRv2 and OLSR excessively flood the network with messages. Therefore, a logical solution to this problem is to modify OLSRv2 by reducing the need for continuous network flooding while maintaining optimum link quality and energy efficiency. The significance of routing techniques in D2D communication systems has pushed many researchers to optimize the routing protocol for efficient and reliable data routing in D2D communication. However, none of the related work presents a structured and energy-efficient link sensing technique, database maintenance technique and multi-criteria mobility, battery, node degree and connection to the base station for MPR selection based on node status, which is the goal of this paper.

The main contributions of the paper are as follows:

- The introduction of a non-forwarding HI and Emergency Hello (EH) messages as a response to HELLO messages and for priority Hello broadcast, respectively.
- A novel message sequence that replaces OLSR's HELLO, HELLO, and TC message sequence with a new sequence that interweaves HELLO and TC messages with three database (DB) chores and LISTEN states for improved link and energy efficiency. The new message sequence is *DB_SELF*, *LISTEN*, *HELLO*, *DB_NEIGHBORS_RCVD*, *HI*, *DB_NEIGHBORS_BOTH*, *TC*, and *DB_ROUTES* sequence.
- The implementation of the modified algorithm into OLSRv2 protocol that selects the best node as an MPR considering the four (4) parameters, namely node battery level (BL), mobility speed (MB), node degree (ND), and connection to base station (CBS) for optimum relay selection, network lifetime, and energy efficiency.

2. Related Work

In wireless devices, a routing protocol operating at the network layer facilitates the transmission of data packets among interconnected nodes [31,32]. In 5G/B5G technology, the implementation of D2D communication requires efficient routing and relay selection to enhance Quality of Service (QoS) performance [24,32]. Gustavo [33] implemented OLSR [15] in the NS-3 simulator by porting OLSR implementation codes from NS2 [34,35]. His implementation added support for MID messages. According to [30], most MANET practitioners believe the loss is caused by MPRs, which tend to function nicely in highly dense environments but create these shortcomings in sparse networks. Others think this is due to the excessive flooding of TC messages in the network [30].

Hamzaoui et al. [36] proposed an improvement of how an OLSR selects an optimal MPR and called it M-OLSR. The improvement incorporates a system whose weight considers density metrics, mobility, and energy. The improvement was dependent on the k-means clustering-enabled mobility metric proposed by Hamzaoui et al. [37]. Their method improved the OLSR algorithm by addressing factors such as adaptability to network conditions, node stability, energy efficiency, predefined thresholds, and expert analysis. This method enhances the versatility of the protocol in optimizing MPR selection across various network scenarios. The dynamic adjustment of the mobility metric within the protocol contributes to the stable and efficient selection of MPR nodes, as well as improving the Quality of Service (QoS) in dynamic network environments. The optimal configuration of this metric which accounts for direction, historical movements, and speed has already demonstrated significant improvements. Future efforts will focus on proposing additional algorithms to further refine the OLSR protocol, reflecting researchers' ongoing dedication to advancing efficient and reliable routing protocols.

Kumari and Sahana [38] proposed a novel algorithm, the Hybrid ACO-PSO Meta-Heuristic (HAPM), which combines Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and a dynamic queue mechanism to enhance Quality of Service (QoS) constraints and reduce data dropping. The HAPM algorithm demonstrated superior performance in key metrics such as throughput, packet delivery ratio (PDR), hop count (Hc), routing overhead, and end-to-end delay, when compared to standalone PSO, ACO, and other hybrid algorithms like the enhanced AODV. Notably, the HAPM algorithm achieved

significant reductions in hop count of 83%, 87%, and 90% compared to the enhanced AODV, PSO, and ACO, respectively. Despite these improvements, the algorithm does not increase time complexity, making it a viable solution for complex network environments. The study also highlighted the growing focus on swarm-based multicast routing protocols, with the HAPM algorithm offering a promising approach for solving multicast routing problems in large networks.

Oyakhire and Gyoda [39] propose an enhanced approach to reduce control packet overhead in dense networks by applying game theory to the optimized OLSR protocol. The method utilizes two policies, “willingness_always” and “willingness_never,” to decrease the MPR ratio by excluding nodes with less influence from being nominated as MPRs. Implemented in Riverbed Modeler 18.0, the improved protocol was compared to the original OLSR and demonstrated superior performance across most metrics, with the exception of a negligible increase in end-to-end delay. The proposed algorithm dynamically adjusts a node’s willingness to become an MPR based on its influence on neighboring nodes, using indexed HELLO messages to prevent duplication. Although the study did not account for the battery power of nodes—an important consideration given OLSR’s high power consumption—future work aims to address this by incorporating battery life and Quality of Service (QoS) parameters, particularly for multimedia applications. Additionally, the authors proposed the implementation of a testbed-enabled improved protocol to further evaluate its efficiency, particularly in real-time, multi-hop network scenarios.

Zhang et al. [40] work on the intrinsic variability and complexity of multi-access edge computing networks by proposing a policy called the Policy Proximal Optimization (PPO). The proposed policy is based on a device-to-device (D2D)-assisted computation offloading and resource allocation scheme. In addressing the challenge of optimizing the utilization of idle computational resources in multi-access edge computing, the researchers implemented ad hoc wireless technology to authenticate network environment mirroring. They tackled the inherent randomness, dynamics, and complexity by modifying the training features of neural networks to generate a deep Q-network on the user’s equipment, aiming to reduce the computation of energy consumption, task execution time, transmission delay, and task count. To achieve this, they formulated a Markov chain decision process designed to minimize energy consumption while reducing time loss, ultimately deriving an optimal offloading strategy. The proposed model focused on achieving these objectives effectively. They did not incorporate an aspect for preserving privacy in the uploading processes of the D2D network.

Maccari and Cigno [41] propose and evaluate two modifications to the MPR selection strategy within the OLSR protocol, addressing the limitations of the standard algorithm, which focuses on local optimality but neglects global network efficiency. The proposed modifications shift the focus from local properties to global optimization, aiming to reduce the overall number of MPR nodes in the network. The first modification minimizes the number of MPRs selected by a node, which in turn reduces the global MPR count and protocol overhead without introducing any additional transmission or computational complexity. This makes it particularly suitable for mobile networks. The second modification, while introducing minimal overhead, significantly enhances route stability within the OLSR framework. Simulations demonstrate that these heuristics can reduce the global number of MPRs by 8–15% and improve routing stability across various network topologies and mobility scenarios. The results indicate that these simple modifications effectively achieve better global network properties, paving the way for future research into more sophisticated heuristics that could further optimize the logical topology created by OLSR.

Zeng et al. [27] addresses the challenge of maintaining Quality of Service (QoS) in Unmanned Aerial Vehicle Ad Hoc Networks (UANETs), where rapid changes in network topology due to UAV movement often lead to increased endwise delay, reduced throughput, and higher packet loss rates. To tackle these issues, the authors propose a Deep Q-Network algorithm which dynamically optimized the link state routing protocol. Their protocol adaptively sends hello messages and topological messages in intervals according to the

speed and position of neighboring in real-time, which improves the routing update capability of nodes. By using DQN, the protocol dynamically adjusts the TC message flooding interval, enabling timely routing table updates that ensure correct data packet transmission even as the network topology rapidly changes. The study demonstrates that DQN-OLSR outperforms the traditional protocols such as AODV, GRP, and standard OLSR in terms of end-to-end delay, packet loss rate, and throughput, especially in random waypoint and random walk mobility models. Furthermore, DQN-OLSR shows strong robustness, maintaining low QoS performance degradation even when positioning errors are introduced. The paper suggests that this approach is particularly effective in environments that showcase quick changes in the network topology. Future work will focus on incorporating additional factors like the link load rate of the MAC layer to further enhance the routing strategy and QoS performance.

To enhance energy efficiency with various mobility speeds of nodes, ref. [42] presents a Disaster Scenario-Optimized Link State Routing (DS-OLSR) protocol with Message Prioritisation (DS-OLSRMP). The DS-OLSRMP is a novel approach to enhance the energy efficiency and communication reliability of disasters in terms of recovery scenarios of low-battery devices using an OLSR protocol. The study introduces the ALERT message, a new type of message designed to prioritize communication based on the battery level of devices, by grouping node priority into four stages, namely Low, Medium, High, and Critical. This proposed solution effectively prioritizes status notifications and message delivery, ensuring that low-battery, energy-consuming devices are not overburdened and can continue to function during critical operations. This strategy includes modifying the multipoint relay (MPR) mechanism to prevent the selection of MPR nodes with low battery, thereby extending their operational lifespan and ensuring timely message delivery. The paper emphasizes the importance of balancing communication efficiency with energy conservation, particularly in the context of disaster response where communication networks must remain operational under challenging conditions. Simulation results demonstrate the effectiveness of the DS-OLSRMP protocol in maintaining high packet delivery rates and reducing the likelihood of message repetition. This contributes to a more resilient and sustainable communication infrastructure for disaster scenarios. The study also evaluates the scalability of the proposed protocol under varying network sizes and node speeds, highlighting its robustness and suitability for large-scale mobile ad hoc networks (MANETs).

Rahul and Kaarthick [43] introduce an innovative protocol to address the energy consumption and routing challenges in Mobile Ad Hoc Networks (MANETs). This is carried out by integrating a chaotic bat swarm with the deep Q-learning Network into the CBS-OLSR protocol, which leverages the MPR technology of OLSR, thereby enhancing it with deep Q-learning to optimize MPR set selection. This results in significantly reducing energy usage while increasing network lifespan. The protocol significantly reduces end-to-end delay and improves packet delivery ratio, addressing the critical issue of controlling packet overhead in existing active routing protocols that often lead to link failures, reduced network bandwidth, and lower Quality of Service (QoS). After the CBS-OLSR model has been integrated with the improved deep Q-learning algorithm, the study demonstrates a substantial enhancement in network performance metrics, including throughput, network lifetime, overhead, and energy consumption, as compared to existing methods. The CBS optimization mitigates the issue of premature convergence by incorporating chaotic functions, which diversify the bat swarm and reduce complexity. Through simulations, the proposed CBS-OLSR protocol shows significant improvements over traditional protocols, offering a more efficient, robust, and sustainable solution for MANETs.

Bunu et al. [44] proposed an optimized approach for device-to-device (D2D) communication to enhance the effectiveness of relay and routing techniques, particularly in Off-Load Traffic scenarios. The paper focuses on addressing the frequent power failures and disconnections in multipoint relay (MPR) nodes within the OLSRv2 protocol, which often results from the selection of unsuitable MPR nodes. To mitigate this issue, a Linear Programming (LP) technique is employed to determine the optimal weight distribution

among four key parameters: battery level, mobility speed, node degree, and connection to the base station. These parameters are crucial in maintaining robust D2D connectivity, and the proposed model ensures that each parameter contributes fairly to the relay selection process. By mathematically modeling these parameters and their influences, the study develops a weighted model that optimizes the selection of relays, thereby improving energy efficiency and overall network Quality of Service (QoS). The LP model is used to calculate the optimal contributions of each parameter, resulting in a more reliable and energy-efficient D2D communication system.

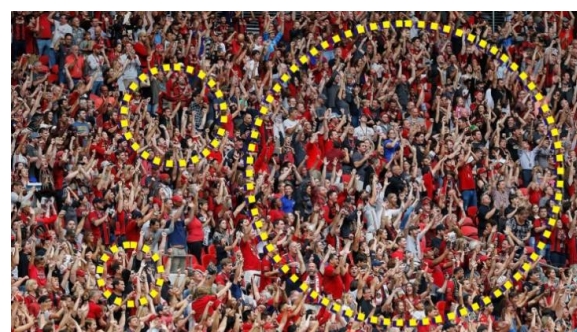
Kazmi et al. [32] provide a comprehensive review of interference mitigation techniques that utilize Software-Defined Networks (SDNs) in wireless communication to explore existing research trends and evaluating existing solutions within the context of B5G networks. The analysis begins with discussing the present perspectives of mobile communication concerning routing-based interference mitigation followed by past and future perspectives. The study then delves into the potential of various SDN topologies—centralized, decentralized, and hybrid for mitigating interference, highlighting their strengths and limitations. By critically analyzing prominent SDN-based interference mitigation schemes, this review identifies gaps in current methodologies and proposes an Intelligent Interference Mitigation architecture for utilizing controllers in distributed SDNs. The review concludes by suggesting future research directions, emphasizing the need for advancements in some areas. These areas include Fog Federation, Lightweight SDN implementations in drone networks, Cognitive Radio Access Networks (C-RANs), 5G Ultra-Reliable Low-Latency Communication (URLLC), and compatible SDN controllers.

3. Differences from Existing Work

3.1. Differences in Message Sequence

The OLSRv2 message sequence mandates MANET/D2D nodes to exchange two HELLO messages before a TC broadcast. The two HELLO messages enable devices to identify their symmetric one-hop and two-hop neighbors and the selection of willing symmetric one-hop neighbors for flooding or routing MPRs.

This paper proposes a different approach to the OLSRv2 message sequence by introducing a HI message as a response to HELLO messages. The proposal equally introduced ample listening periods to enable the controlled completion of database chores. It is imperative to allow nodes to complete their link state commitment to the database if OLSRv2 is to operate as desired. While OLSRv2 mandates one-hop connectivity between nodes, in practice, the one-hop neighbors of related devices that are equipped with powerful transceivers could be in their hundreds or thousands, for example, in a packed stadium, as shown in Figure 2. Saving thousands of reported one-hop neighbors to OLSRv2 information bases using OLSRv2 existing message intervals will be problematic for OLSRv2 because frequent control messages broadcast will disrupt the process.



■■■■ Radio broadcast range of MANET transceivers determines the number of 1-hop neighbors

Figure 2. Stadium scenario. The number of one-hop neighbors reported by OLSRv2 is determined by the broadcast range of transceivers on the various MANET devices owned by users.

3.2. Enforcement of DB Chores Period

The second difference between the current specification of OLSRv2 and the proposed modification is the introduction of DB chore periods. Nodes are mandated to operate in listening mode while information-based operations are ongoing. In the DB chores state, nodes are permitted to discard received broadcasts to prevent silently disrupting database maintenance operations.

3.3. Suitable MPR Selection Techniques

Another critical difference between the existing research on OLSRv2 and this research is the ability to utilize four (4) factors—the battery level of the nodes (BL), mobility speed (MB), the degree of nodes (ND), and connection to the base station (CBS)—to choose relays in the best possible way, lengthen the network's lifespan, and use less energy. The proposed MPR selection mechanism combines these four parameters into one metric to reduce energy dissipation and control routing overhead due to transmit each parameter individually.

4. System Design and Goals

This section presents the proposed design goals, the new message type, message sequence, HI message packet format, and a new technique for the MPR selection process. The current design specification for OLSRv2 [45] and OLSR [15] assumes mobile devices (such as mobile phones) have abundant redundant processing and storage capacity to handle frequent link- and route-building broadcasts/processing and the housekeeping or maintenance of their associated information bases. Unfortunately, this is not often the case. Therefore, this paper acknowledges that mobile devices have limited resources for OLSRv2 link sensing and information-based maintenance services. Thus, the proposed design emphasizes controlled broadcasts and ample periods for proper information base maintenance. These design goals will help to improve the OLSRv2 performance in link sensing, the MPR selection mechanism, and database maintenance techniques that will produce desirable results in dense and sparse static and/or mobile network scenarios.

The primary design goals of this paper with respect to enhancing the OLSRv2 packet delivery ratio, reducing network energy consumption, and maintaining up-to-date information bases are as follows:

- Network decongestion: OLSRv2 implementation is plagued by enormous packet loss and high processor loads due to uncontrolled broadcasts of control messages. However, enforcing listening periods before and between message broadcasts will reduce the frequent usage of the radio transmitter since nodes that transmit HELLO messages and fail to receive a timely response (due to an ongoing DB chores mode) will simply drop into listening mode. The reduced usage of the radio transmitter increases the energy available to the D2D nodes since transmitting consumes more energy than receiving.
- Information base maintenance: the proposed database (DB) chores mode allows OLSRv2 to commit link states and routes to information bases without sudden disruption from incoming messages on the OLSR address.
- MPR selection: the proposed MPR selection considers the status of the nodes to select the best node as the MPR for energy efficiency and link quality.

4.1. Message Format

The new HI message type conforms to the MANET neighborhood discovery protocol (NHDP) specification [13]; thus, the HI message is technically similar to the HELLO message, with the following exceptions:

- When the HI message type is the HI message instead of the HELLO message.
- When the HI message is only generated in response to the HELLO message.
- When nodes are not permitted to generate a HI message without first hearing the initial HELLO message.

- When only nodes that generated the initial HELLO message can respond to a HI message after the DB chores period.

The only authorized response to a HI message after the DB period is with a TC message.

4.2. Message Sequence

The proposed message sequence is designed to decongest OLSRv2 message broadcasts and enhance information base maintenance. The new message sequence can achieve the dual goals of network decongestion and optimum information base maintenance by ensuring nodes refrain from responding to OLSRv2 message broadcasts while writing to OLSRv2 tables. Although it is called a message sequence, the actual messages are preceded by either a variant of DB_SELF or LISTEN. The proposed message sequence is DB_SELF, LISTEN, HELLO, DB_NIEGHBORS_RCVR, HI, DB_NIEGHBORS_BOTH, TC, DB_ROUTES.

4.2.1. DB_SELF State

This sequence occurs if:

- The router starts up for the first time. It records all its interface addresses in the Local Interface Set [46]. MANET/D2D or OLSR addresses are marked with a 1, while non-OLSR addresses are marked with a 0.
- The address of the router changes, and it records the information in the Removed Interface Address Set [46].

Nodes in the DB_SELF state can immediately move to the DB_NEIGHBORS_RCVD state if a HELLO broadcast is received on their OLSR interface, and if they are finished with updating the content of their Local or Removed Interface Sets. This state is shown in Figure 3.

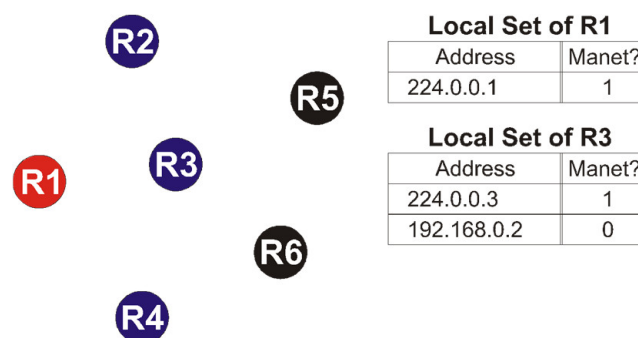


Figure 3. Group of nodes with sample Local Set entries.

4.2.2. LISTEN State

The LISTEN state or period can be used for the following:

- To compute network traffic by listening to the ongoing HI and TC messages broadcast. Network traffic computation can be used to determine the duration of the next LISTEN state or the next DB_CHORE state. Such states can be increased if the network traffic is heavy or set to default if the network traffic is light.
- To create or update the Link and Neighbor Sets of the node from received HI and TC messages.

This state can be quickly exited if:

- A HELLO broadcast is heard.

Nodes in the LISTEN state can immediately move to the next state (DB_NEIGHBORS_RCVD) if a HELLO broadcast is received. The state is summarized in Figure 4 below.

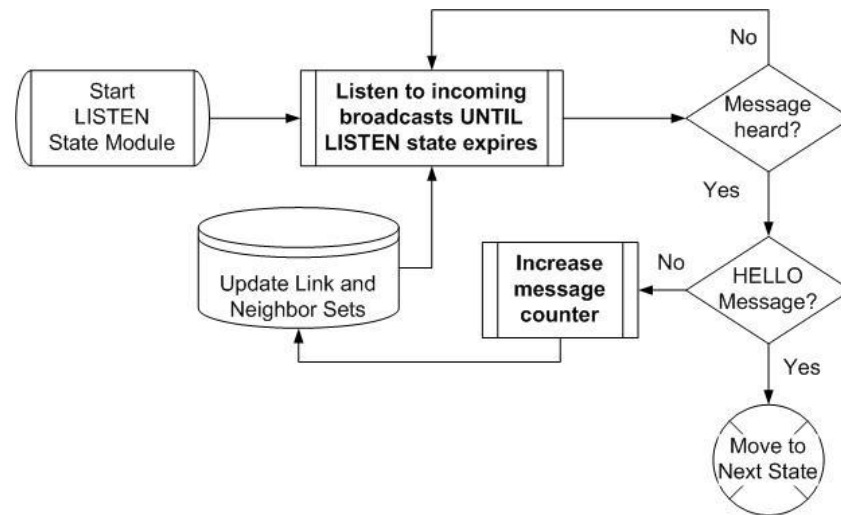


Figure 4. LISTEN state process.

4.2.3. HELLO Message State

Nodes use this state for the following:

- To send a HELLO broadcast. HELLO broadcasts can be sent if any of the following conditions are met:
- At the expiration of the LISTEN state
- At the expiration of DB_SELF and before the LISTEN state if the EMERGENCY_HELLO constant is greater than zero.

Nodes that emitted the HELLO message must immediately set EMERGENCY_HELLO to zero (default value is -1) and await a HI response or wait for the expiration of DB_NEIGHBORS_RCVD (see flowchart in Figure 5).

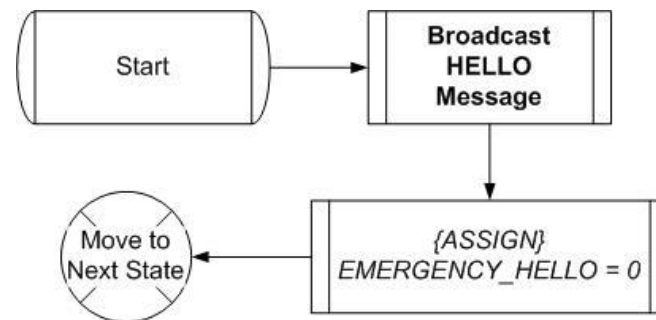


Figure 5. HELLO message process.

4.2.4. DB_NEIGHBORS_RCVD State

Nodes that received or heard the HELLO broadcast will use this state for the following:

To update their Link and Neighbor Sets with the OLSR addresses of all nodes that emitted HELLO messages (see Figure 4).

Figure 6 depicts Neighbor Set entries created by the three nodes (blue, R2–R4) that received or heard the HELLO broadcast of R1. The last two nodes (denoted in black) are outside the range of R1’s HELLO broadcast, hence their Link and Neighbor Sets are empty. The value of the HELLO message is the OLSR address of the sender, which is included in the Link and Neighbor Sets. The symmetric status of the link is initially set to 0 since it is not certain if R1 is within the range of their individual broadcast.

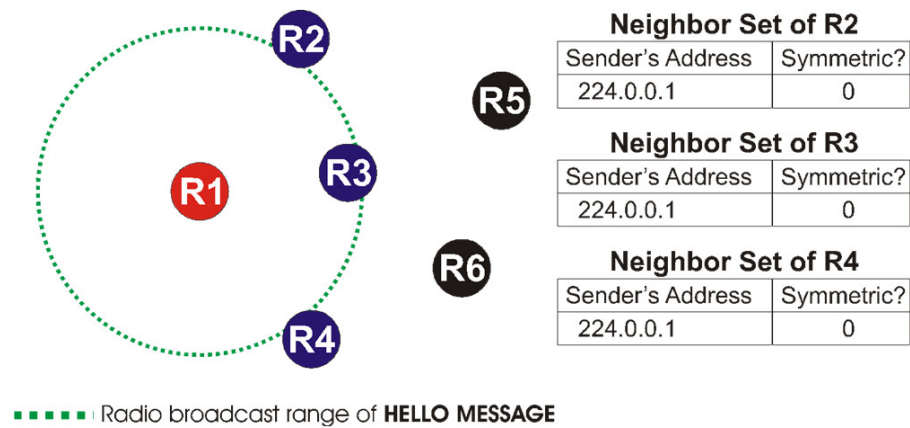


Figure 6. Neighbor Set after HELLO broadcast.

4.2.5. HI Message State

Nodes that received or heard a HELLO broadcast will use this state to:

Broadcast a HI message with the addresses of all nodes whose HELLO messages were received, as shown in Figure 7.

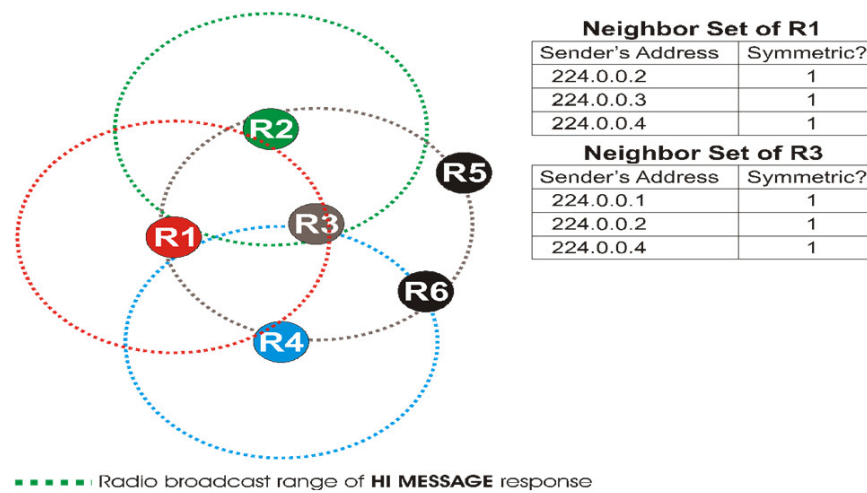


Figure 7. HI message broadcast and sample Neighbor Set after broadcast.

- Here, EMERGENCY_HELLO is set to 1 or a value that is greater than zero. This will enable the node to broadcast a HELLO message before the next LISTEN state. Nodes that broadcasted the initial HELLO message are mandated to equally broadcast a HI message for the following reasons:
- To determine the symmetric connection between nodes.
- To prevent a break in the message sequence; for example, when all nodes broadcast a HELLO message concurrently, all nodes must therefore respond with a HI message to maintain the message sequence.

The three nodes (green, brown, and cyan) that heard or received the initial HELLO broadcast responded with a HI message after the expiration of the DB_NIEGHBORS_RCVD state. Broadcasts from the three nodes are received by both the node that initiated the HELLO message (red) and the nodes that were outside the range of the initial HELLO broadcast (black). Nodes that received or heard the HI message will update their Link and Neighbor Sets. R1 will set the symmetric status of links R2–R4 to 1 since they all responded to its initial HELLO message, while nodes R5 and R6 will set the symmetric status of links R2–R4 to 0 (asymmetric) because the ability of those nodes to respond to the broadcasts is still uncertain.

Any node that did not broadcast a HELLO message but received a HI message must do the following:

- Set EMERGENCY_HELLO to 1 or a value that is greater than zero. This will enable the node to broadcast a HELLO message before the next LISTEN state. The HI message process or state is summarized in the flowchart in Figures 8 and 9 below.

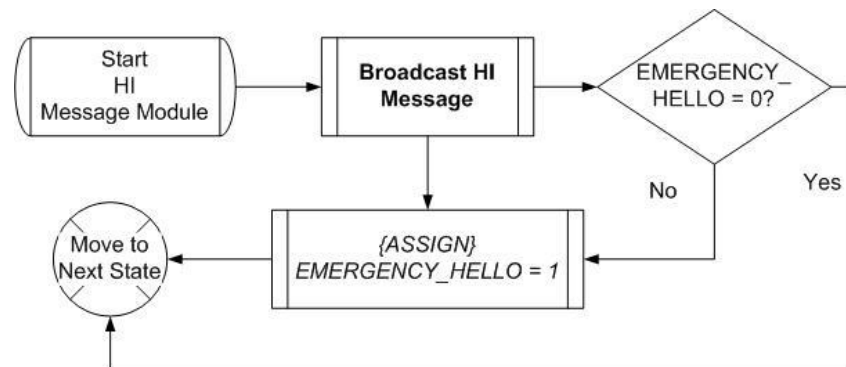


Figure 8. HI message process for sender.

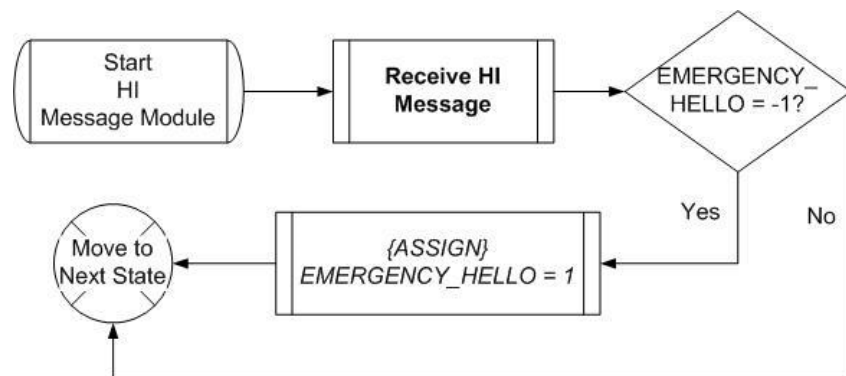


Figure 9. HI message process for receiver. Receivers who did not hear initial HELLO message must set EMERGENCY_HELLO to 0.

4.2.6. DB_NEIGHBORS_BOTH State

DB_NEIGHBORS BOTH state expects nodes to enact the following things:

- Update their Link, Neighbor, and two-hop sets with the OLSR addresses of all nodes that participated in the HELLO/HI broadcast (see Figures 4 and 5).
- Select MPRs from the list of one-hop neighbors based on the OLSRv2 MPR selection process.

4.2.7. TC Message State

This research proposes two conditions that will determine if TC messages should be emitted by an MPR; these are as follows:

- The availability of symmetric two-hop neighbors between the MPR and its selectors.
- Changes in the OLSR addresses of the MPR selectors.
- Changes in the number of nodes that selected the MPR to advertise their links.

4.2.8. DB_ROUTES State

The final state in the proposed research is the DB_ROUTES. Nodes across the network use the advertised links extracted from TC messages to populate the following tables or sets:

- Advertising Remote Router Set;

- Router Topology Set;
- Routable Address Topology Set;
- Attached Network Set.

DB_ROUTES has the longest TTL amongst the proposed message sequence. The proposed TTL for DB_ROUTES is 10 s per MPR group.

HI Message Packet Format

The new HI message packet format conforms to the generalized MANET packet format as specified in [47]. The general format of a HI message is shown in Figure 10.

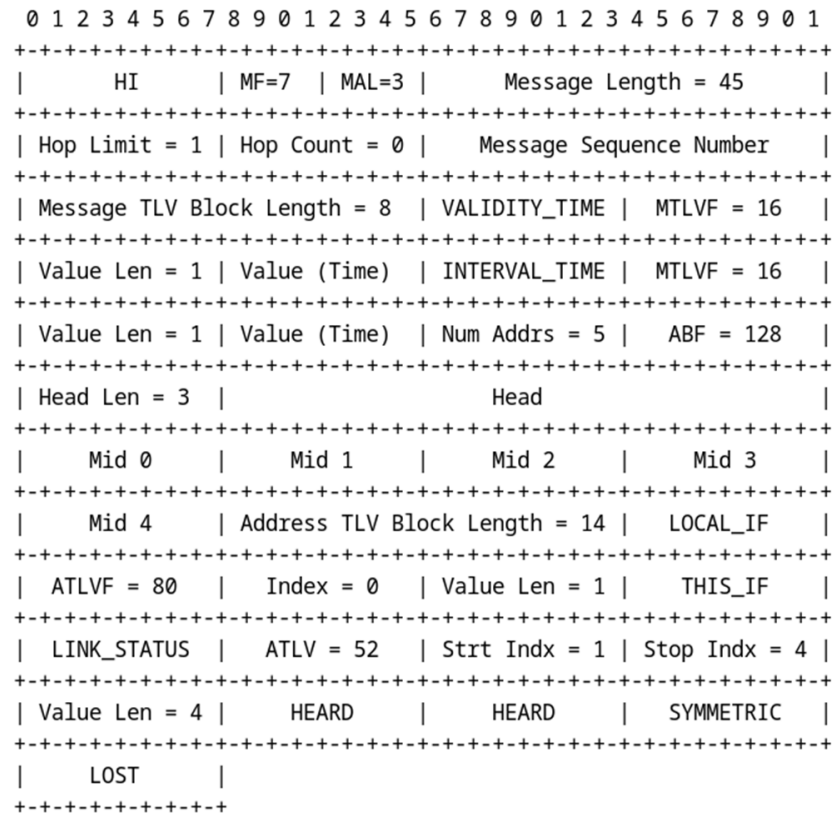


Figure 10. HI packet format; note that the message type is HI, but the format is similar to the HELLO packet [46].

4.3. Protocol Parameters and Constants

When designing and implementing wireless network protocols, particularly for D2D communication networks, specific parameters and constants are usually defined to control the behavior and performance of the protocol. These parameters and constants can vary depending on the type of protocol and specific goals of the protocol design. These parameters and constants equally played an important role in determining the behavior and performance of wireless network protocols in simulations, and testbed and real deployments. When simulating a D2D wireless network protocol, it is vital to carefully choose and tune these parameters to reflect realistic operating conditions and to ensure accurate evaluation of the protocol’s performance. This paper proposes new protocol parameters with their constants and also proposes the modifications to existing parameters and constants defined in [46,47] for efficient energy consumption and link management. These parameters and constants are of primary importance for tuning the behavior and performance of routing protocols in various network environments. The specific values chosen for these parameters can significantly impact the efficiency, reliability, and scalability of our proposed routing protocol.

4.4. Modification to Message Intervals and New Message Variables

In wireless routing protocols, message intervals and message variables play a critical role in determining how efficiently and reliably the protocol operates. These elements are crucial for maintaining up-to-date routing information, detecting changes in the network topology, and ensuring timely transmission between nodes in a network. This paper proposed new message variables that regulate the control message broadcast interval in the proposed routing protocol. These message intervals and new message variables are crucial for the proper functioning and performance optimization of routing protocols in wireless networks. They help to ensure that the network can adapt to dynamic changes, maintain accurate routing information, and provide efficient data forwarding. The new variables under this parameter are as follows:

- **DB_SELF_INTERVAL (DSI);**
- **DB_NIEGHORS_RCVR_INTERVAL (DNRI);**
- **DB_NIEGHORS_BOTH_INTERVAL (DNBI);**
- **DB_ROUTES_INTERVAL (DRI);**
- **EMERGENCY_HELLO (EH);**
- **LISTEN_MIN_INTERVAL (LMI);**
- **LISTEN_INTERVAL (LI).**

DB_SELF_INTERVAL (DSI) is the state where D2D communication nodes are handling their own database or internal data structures, and is related to maintaining local information about itself, such as its interfaces, neighbors, or routing tables. The default time in seconds proposed for nodes to update their Local Set is 2 s. Therefore, $DSI = 2$.

DB_NIEGHORS_RCVR_INTERVAL (DNRI) is the state where a node has received information about its neighbors, typically through a HELLO or TC messages. In this state, the node might be updating its internal database or data structures with the newly received neighbor information. The DNRI interval is crucial for maintaining up-to-date and accurate routing information, particularly in dynamic networks where neighbor relationships can change frequently. The default time in seconds proposed for nodes to update their Link and Neighbor Sets is 5 s. Therefore, $DNRI = 5$.

DB_NIEGHORS_BOTH_INTERVAL (DNBI): **DB_NIEGHORS_BOTH** is a state or condition where a node maintains information about neighbors that have a bidirectional relationship. In networking, bidirectional (or symmetric) links are those where two nodes can both send and receive messages to/from each other successfully. This is important for certain routing protocols that require bidirectional communication between neighbors to establish reliable routes. The **DB_NEIGHBORS_BOTH_INTERVAL** refers to the interval at which a node in an update or network verifies the status of its symmetric neighbor relationships. This interval is used to ensure that the information about neighbors is up-to-date and reflects the current state of the D2D communication network. The default time in seconds proposed for nodes to update their Link and Neighbor Sets is 5 s. Therefore, $DNBI = 5$.

DB_ROUTES_INTERVAL (DRI): Nodes across the network use advertised links extracted from TC messages to populate and update their link and Neighbor sets. The default time in seconds proposed for nodes to update their Link and Neighbor Sets is 10 s per group of symmetric one-hop nodes.

EMERGENCY_HELLO (EH): This variable determines which node or group of nodes has priority HELLO broadcast. Priority HELLO broadcast occurs after the expiration of the DSI. Nodes that broadcasted HELLO must set EH to 0; nodes that did not emit HELLO but emitted HI or heard the HI message without emitting a HELLO message must set EH to 1. Other variables that are important for the efficient performance of our proposed routing protocol are **LISTEN_MIN_INTERVAL (LMI)** and **LISTEN_INTERVAL (LI)**. LMI represents the minimum time before the transmission of the HELLO message by a D2D communication node with a default LMI of 2 s, while LI represents the maximum time before the transmission of a HELLO message by a D2D communication node. Both LMI and LI can be exited early if the node in this state hears a HELLO message. The default

LI = 3. The processing techniques of the proposed routing protocol have been explained in the following algorithms.

4.4.1. DSI Function

This module retrieves the addresses of all network interfaces on a D2D communication node and saves them to the OLSRV2 Local Set. It uses the *IsD2D* variable to determine if the address was assigned by OLSR. Nodes that responded to HELLO messages with a HI message and have not broadcasted a HELLO message can broadcast their HELLO messages once interface data are written to the Local Set—only if EH = 1. The procedure for the DSI is explained in Algorithm 1. This algorithm is designed to manage the state transitions of nodes in a D2D communication network, particularly focusing on the “DB_SELF” state. The goal is to optimize routing performance by ensuring that each node efficiently updates and maintains its routing information based on its interactions with neighboring nodes. The “DB_SELF INTERVAL” is a critical parameter that defines how frequently a node checks for incoming HELLO messages and updates its state. The interval is set based on the network’s dynamics, such as node mobility and topology changes, to ensure timely updates without causing unnecessary overhead.

Algorithm 1: DSI Function

```

1. DSIModule()
2. For  $\forall$  Address  $\in$  IPAddress[] do:
3.   IsD2D  $\leftarrow$   $\begin{cases} 1 & \text{if } GetType(A) \leftarrow D2D \\ 0 & \text{otherwise} \end{cases}$ 
4.   if IsD2D  $\leftarrow$  1, then:
5.     OLSRAddr = Address
6.     SaveRecord(Address, IsManet, Local)
7.   End For
8.   DSLTimer.Enable  $\leftarrow$  0
9.   if EH  $\leftarrow$  1, then:
10.    BroadcastMessage(“HELLO”, OLSRAddr)
11.   LI.Timer.Enable  $\leftarrow$  True

```

4.4.2. LI Function

This module captures all incoming packets and either discards them (if the packet’s validity time (VTime) has expired) or it extracts and stores key information from the packet into arrays for further processing (that is, saving array information into Neighbor and Link Sets where applicable). The module terminates if either the HELLO packet is received or if the LI timer expires. The algorithm broadcasts a HELLO message at the end of the LISTEN period, only if EH = 1 or if EH = -1 and no HELLO message was heard. The procedure for LI is explained in Algorithm 2. The algorithm is designed to enhance routing efficiency in our D2D network by leveraging the **LISTEN state**. This state allows a node to monitor network traffic and update its Link and Neighbor Sets based on received HELLO (HI) and Topology Control (TC) messages. The algorithm dynamically adjusts the duration of the LISTEN state or the subsequent DB_CHORE state based on the observed network traffic. Each node enters the **LISTEN state**, where it passively listens to network traffic by receiving HI and TC messages from neighboring nodes. During the LISTEN state, the node computes the level of network traffic based on the frequency and density of the received HI and TC messages. After completing the LISTEN state, the node may transition to the **DB_CHORE state**, where it performs tasks such as route calculation, database maintenance, or message generation based on the updated Link and Neighbor Sets.

Algorithm 2: LI Function

```

1. Function LI_Module()
2.   AddressArray ← []
3.   i ← 0, HelloHeard ← 0
4.   While LI.TimerIsActive do:
5.     Packets ← "", Address ← ""
6.     Try:
7.       Address ← getOLSRHost()
8.       Socket ← (Address, 269)
9.       Packet ← getPacketStream(Socket)
10.      Socket.Close()
11.     Catch all exceptions and discard silently:
12.     For  $\forall$  CurrentPacket  $\in$  Packets, do:
13.       MType ← GetMessType(CurrentPacket)
14.       VTime ← GetVTime(CurrentPacket)
15.       Address[] ← GetAddr(CurrentPacket)
16.       if (MType ← "HELLO" || MType ← "HI") & (VTime  $\leq$  HeHiHoldTime)
17.         AddressArray[i][i] ← Addresses
18.         MTypeArray[ArrayCounter] ← MType
19.         HelloHeard ← 1
20.       if HelloHeard ← 1, breakLoop : else i ← i + 1
21.     End While
22.     LI.Timer.Enable ← False
23.     if EH ← 1 || EH ← -1, then:
24.       BroadcastMessage("HELLO", OLSRAddr)
25.       EH ← 0
26.     DNRI.Timer.Enable ← True

```

4.4.3. DNRI Function

This function iterates through the values that were stored in the three array variables (MTypeArray, AddressArray, and HoldTimeArray) by Algorithm 2. Each group of related records (comprising message type, hold time, and addresses) is saved or updated to Link and Neighbor Sets. The algorithm broadcasts the HI message at the end of the database update and sets EH to 1 if the node has not previously emitted a HELLO message. Setting EH to 1 allows the nodes to broadcast a HELLO message before the next LISTEN period (LI function).

The process, as presented in Algorithm 3, is designed to optimize routing in our proposed protocol by effectively managing the **network topology**. This interval governs how frequently a node processes HELLO messages to update its Neighbor Set, ensuring that the node maintains accurate and timely information about its immediate network environment. Each node starts by entering the **DB_NEIGHBORS_RCVD** state upon receiving a HELLO message from a neighboring node. This state is critical for processing neighbor information and updating the node's Neighbor Set. The node examines the HELLO message to extract information about its neighbors, such as their identities, link statuses, and other relevant parameters.

Algorithm 3: DNRI Function

```

1. DNRI_Module()
2.   i ← 0
3.   For  $\forall$  PacketType  $\in$  MTypeArray, do:
4.     For  $\forall$  Addr  $\in$  AddressArray[i][i], do:
5.       SaveRecord(Addr, 0, "Link")
6.       SaveRecord(Addr, 0, "Neighbor")
7.       UpdateSymmetric(Addr, "Link")
8.       UpdateSymmetric(Addr, "Neighbor")
9.     i ← i + 1
10.  End For
11.  DNRI.Timer.Enable ← 0
12.  BroadcastMessage("HI", OLSRAddr)
13.  If EH ← -1, then:
14.    EH ← 1
15.  DNRI.Timer.Enable ← 1

```

4.4.4. DNBI Function

The major difference between the DNRI and DNBI function is that DNRI only processes HI messages and uses the information extracted from HI packets to determine one-hop and two-hop symmetric neighbors. MPR selection processes also take place during this period since each node has enough information to determine nodes with which they share a two-hop symmetric neighbor. Nodes are expected to spend some part of the DNBI period processing incoming HI messages and building links from such messages before committing them to tables. Technically, Algorithm Six is a combination of Algorithms Four and Five; however, Algorithm Six checks if there are changes in the Link and Neighbor Sets before broadcasting TC messages. The procedure is presented in Algorithm 4. The algorithm is designed to optimize the frequency at which a node updates both its Neighbor Set and Link Set in a wireless ad hoc network. This interval dictates how often the node processes incoming HELLO messages to maintain accurate and up-to-date information about its immediate network environment. The node uses the DB_NEIGHBORS INTERVAL to determine when to process HELLO messages. These messages provide critical information about neighboring nodes and the status of direct links between the node and its neighbors. During each interval, the node updates both its **Neighbor Set** (tracking all recognized neighbors) and its **Link Set** (monitoring the quality and status of connections).

Algorithm 4: DNBI Function

```

1. Function DNBI_Module()
2.   AddressArray  $\leftarrow$  [],  $i \leftarrow 0$ , HelloHeard  $\leftarrow$  0, StartTime  $\leftarrow$  Time.Now()
3.   While Time.Now() - StartTime < 3 do:
4.     Packets  $\leftarrow$  "", Address  $\leftarrow$  ""
5.     Try:
6.       Address  $\leftarrow$  getOLSRHost()
7.       Socket  $\leftarrow$  (Address, 269)
8.       Packets  $\leftarrow$  GetPacketStream(Socket)
9.       Close Socket
10.      For  $\forall$  CurrentPacket  $\in$  Packets, do
11.        Address  $\leftarrow$  getOLSRHost()
12.        Socket  $\leftarrow$  (Address, 269)
13.        Packet  $\leftarrow$  getPacketStream(Socket)
14.        Socket.Close()
15.      Catch all exceptions and discard silently:
16.      For  $\forall$  CurrentPacket  $\in$  Packets, do:
17.        MType  $\leftarrow$  GetMessType(CurrentPacket)
18.        VTime  $\leftarrow$  GetVTime(CurrentPacket)
19.        Address[]  $\leftarrow$  GetAddr(CurrentPacket)
20.        IF MType  $\leftarrow$  "Hi" & VTime  $\leq$  HeHiHoldTime:
21.          AddressArray[i][i]  $\leftarrow$  Addresses
22.          MTypeArray[ArrayCounter]  $\leftarrow$  MType
23.          HelloHeard  $\leftarrow$  1
24.         $i \leftarrow i + 1$ 
25.      End While
26.       $i \leftarrow 0$ , CEx  $\leftarrow$  0#Change Exists
27.      For  $\forall$  PacketType  $\in$  MTypeArray, do:
28.        For  $\forall$  Addr  $\in$  AddressArray[i][], do:
29.          CEx  $\leftarrow$  CompareRecord(Addr, 0, "Link")
30.          CEx  $\leftarrow$  CompareRecord(Addr, 0, "Neighbor")
31.        IF CEx  $\leftarrow$  1 : Break Loop, Else  $i \leftarrow i + 1$ 
32.      End For
33.      IF CEx  $\leftarrow$  0 :  $i \leftarrow 0$ 
34.      For  $\forall$  Addr  $\in$  AddressArray[i][i], do:

```

Algorithm 4: Cont.

```

35.           For  $\forall$  Addr  $\in$  AddressArray[i][i], do:
36.             SaveRecord(Addr, 0, "Link")
37.             SaveRecord(Addr, 0, "Neighbor")
38.             UpdateSymmetric(Addr, "Link")
39.             UpdateSymmetric(Addr, "Neighbor")
40.            $i \leftarrow i + 1$ 
41.           SelectMPR()
42.           BuildAdLinks()
43.           BroadcastMessage('TC', OLSRAddr)
44.           TcSent  $\leftarrow$  1
45.           DNBI.Timer.Enable  $\leftarrow$  0
46.           DRI.Timer.Enable  $\leftarrow$  1
47.       Else IF TcSent  $\leftarrow$  1:
48.           ExtendTTL()

```

4.4.5. DRI Function

The **DRI_Module** function is designed to process incoming proposed routing protocol packets, specifically focusing on identifying and handling Topology Control (TC) messages. Algorithm 5 extracts relevant address information from these messages and stores them in various routing-related records for future use. Nodes are expected to spend some part of the DRI period processing incoming TC messages and building links from such messages before committing them to tables. The algorithm begins by initializing an empty AddressArray and setting an index i to zero. A variable HelloHeard is initialized to zero to track whether any TC messages have been heard during the execution. The StartTime is recorded to limit the execution time of the main loop to 4 s. The **DRI_Module** processes packets, focusing on TC messages to update routing records. By running within a set time limit and handling potential errors gracefully, the algorithm ensures that critical routing information is accurately recorded and available for future network operations.

Algorithm 5: DRI Function

```

1. Function DRI_Module()
2.   AddressArray  $\leftarrow$  [[]],  $i \leftarrow$  0, HelloHeard  $\leftarrow$  0, StartTime  $\leftarrow$  Time.Now()
3.   While Time.Now() - StartTime < 4 do:
4.     Packets  $\leftarrow$  "", Address  $\leftarrow$  ""
5.     Try:
6.       Address  $\leftarrow$  getOLSRHost()
7.       Socket  $\leftarrow$  (Address, 269)
8.       Packets  $\leftarrow$  GetPacketStream(Socket)
9.       Close Socket
10.      For  $\forall$  CurrentPacket  $\in$  Packets, do
11.        Address  $\leftarrow$  getOLSRHost()
12.        Socket  $\leftarrow$  (Address, 269)
13.        Packet  $\leftarrow$  getPacketStream(Socket)
14.        Socket.Close()
15.      Catch all exceptions and discard silently:
16.      For  $\forall$  CurrentPacket  $\in$  Packetsdo:
17.        MType  $\leftarrow$  GetMessType(CurrentPacket)
18.        VTime  $\leftarrow$  GetVTime(CurrentPacket)
19.        Address[]  $\leftarrow$  GetAddr(CurrentPacket)
20.      End For
21.      IF MType  $\leftarrow$  "TC" & VTime  $\leq$  HeHiHoldTime then:
22.        AddressArray[i][i]  $\leftarrow$  Addresses
23.        MTypeArray[ArrayCounter]  $\leftarrow$  MType

```

Algorithm 5: Cont.

```

24.         HelloHeard  $\leftarrow$  1
25.          $i \leftarrow i + 1$ 
26.     End While
27.      $i \leftarrow 0$ 
28.     For  $\forall$  PacketType  $\in$  MTypeArray, do:
29.         For  $\forall$  Addr  $\in$  AddressArray[i][], do:
30.             SaveRecord(Addr, 0, "AdvertRemoteRouter")
31.             SaveRecord(Addr, 0, "RouterTopology")
32.             UpdateSymmetric(Addr, 0, "RoutableAddrTopology")
33.             UpdateSymmetric(Addr, 0, "AttachedNetwork")
34.         End For
35.      $i \leftarrow 0$ 
36. End For
37. DRI.Timer.Enable  $\leftarrow$  False
38. DSI.Timer.Enable  $\leftarrow$  True

```

The LMI_Module algorithm is not presented since it is essentially a copy of the LI_Module, but with a lower LISTEN duration. The following are suggested as deployment scenarios for the LMI_Module:

- Where the number of nodes is known, the listening period can be as small as possible.
- Where the number of one-hop nodes can be dynamically determined during the longer LI_Module (by counting the number of detected message types), the implementation can switch to the LMI_Module or reduce the LI value if the number of messages from one-hop nodes is small.

The proposed entry point of the OLSRv2 modification is the main function. Thereafter, each algorithm is executed as shown until control is returned to Algorithm 1. In general terms, the algorithms discussed provide robust mechanisms for our proposed routing protocol by effectively managing various states, intervals, and message processing functions. By dynamically adjusting state durations like the LISTEN and DB_NEIGHBORS_BOTH intervals, the algorithms ensure that nodes can adapt to changing network conditions, thereby improving the accuracy of routing decisions and conserving resources in both dynamic and stable environments. The DRI_Module algorithm demonstrates a focused approach to processing our proposed protocol packets, especially Topology Control (TC) messages. By extracting and storing crucial address information within a time-bound window, it enhances the efficiency and reliability of the network's routing functions. With built-in resilience through exception handling and comprehensive record-keeping, the algorithm ensures that nodes maintain an up-to-date understanding of the network topology, which is essential for maintaining optimal routing performance.

Together, these algorithms represent a cohesive approach to managing routing in D2D communication networks, balancing the need for timely updates with resource efficiency, and providing a solid foundation for reliable and scalable network operations.

4.5. MPR Selection Technique

The proposed MPR selection mechanism allows a node to consider four (4) parameters, namely node battery level (BL), mobility speed (MB), node degree (ND), and connection to the base station (CBS) for optimum relay selection, extended network lifetime, and energy efficiency, as presented in our previous work in [48]. The mechanism classified nodes into various node statuses, namely best, better, good, and bad, and returns a single value of 1, 2, 3, or 4, respectively, in OLSRv2 Hello and TC messages, thereby saving energy and other resources required for much topological information. The modified Hello and TC message Type Length Value (TLV) is presented in Figure 11, and the algorithm for MPR selection is shown in Figure 12.

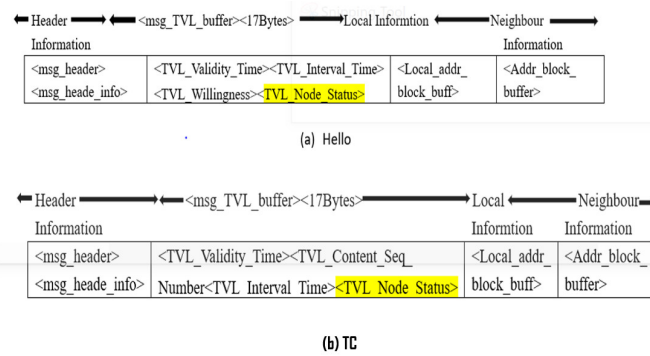


Figure 11. The modified Hello and TC message TLV for the proposed routing protocol.

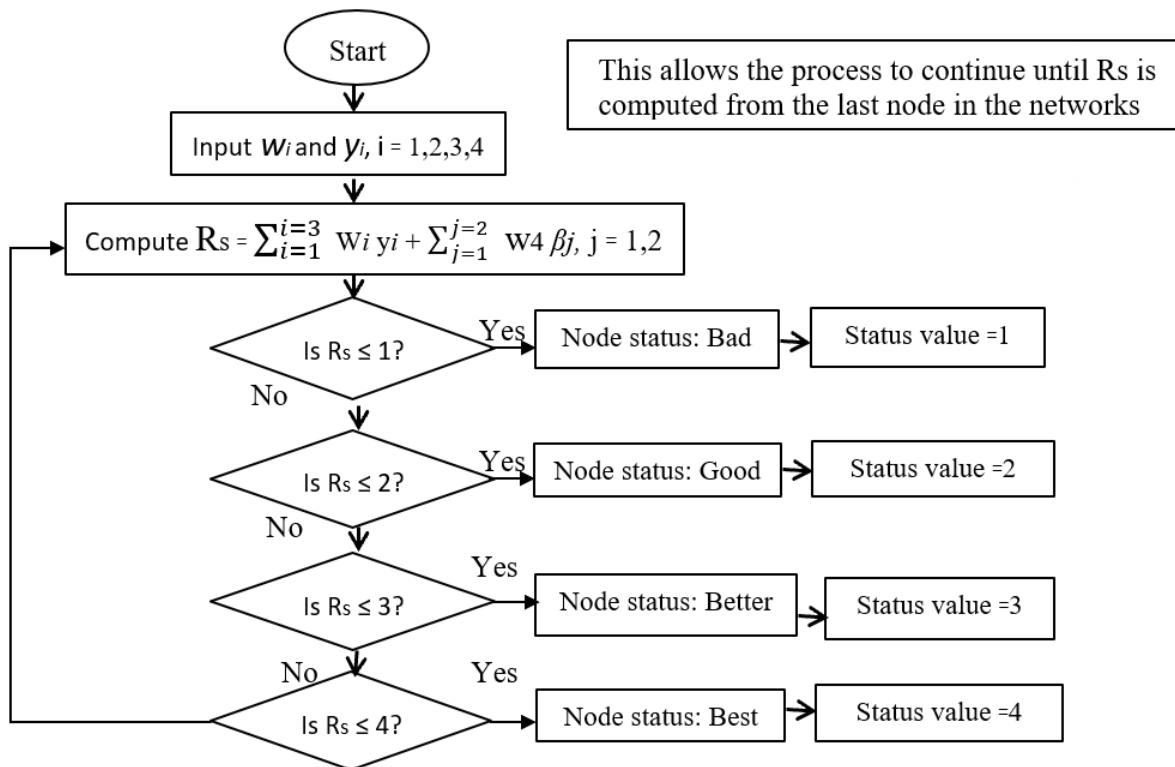


Figure 12. A flow chart of the relay selection process for a D2D network, showing the node status depending on the relay selection model [44].

4.5.1. Algorithm for MPR Selection

This section gives the algorithm for the MPR selection, which depends on the weights of four parameters, namely battery level, mobility speed, node degree, and connection to the base station. The classification of the model parameters and the relay selection model is given in our previous papers [43,48]. The MPR selection utilizes node status, as illustrated in Figure 12 above.

4.5.2. MPR Selection Algorithm

The proposed selection of the MPR process has been optimized to utilize node status, thereby allowing the best node to be selected as the MPR. The MPR selection is modified from the conventional MPR selection in the OLSRv2 protocol. The MPR selection is represented by the value of the node status that determines the best node as the relay. This is given in Algorithm 6, where the willingness level takes four possibilities, namely WILL NEVER, with a status value of '1', WILL LOW, with a status value of '2', WILL DEAFULT, with a status value of '3', and, finally, WILL HIGH, with a node status of '4'.

The proposed MPR selection algorithm ranked willingness based on the node's status. The low-status value of 1 indicates that the node status is bad and will not be selected as the MPR. Meanwhile, a high status value of 4 indicates that the node is the best one to be selected as the MPR and that willingness is high. Therefore, the willingness increases with the increase in the status value. Each node in the network elects its MPR during the MPR selection process from its one-hop neighbor through the advertisement of its willingness, which is broadcasted via HELLO message. Therefore, by implementing this algorithm, the best high-status nodes will be considered as the MPR first, followed by those classed as better and then good. Thus, low-status nodes will be considered last, thereby extending their lifetime. This reduces transitional route disappearances to the other edge of the network, which can lead to network collisions, thereby increasing control overhead in the network.

Algorithm 6: MPR Willingness Process

Begin:

Require: NodeStatus > NS_l

Ensure: Appropriate NodeStatus && Two-hop Nodes

NS_h ← the highest node's status

NS_l ← the lowest node's status

NS_c(i) ← the present status of node i

WL_i ← willingness of node i

Obtain NS_c(i)

If NS_c(i) < NS_l then

WL_i = WILL NEVER → //Bad

Else if NS_c(i) = 2

WL_i = WILL LOW → //Good

Else if NS_c(i) = 3

WL_i = WILL DEFAULT → //Better

Else

WL_i = WIL HIGH → //Best

End if

Return Willingness level

5. Network Model

5.1. Mobility Scenario

A node's mobility in a D2D communication network can be modeled based on two primary parameters: speed and pause time. Mobility in any wireless network setting leads to alterations in topology, heightened routing overhead, and increased link failure rates. The most used mobility model for evaluating wireless networks is the random waypoint (RWP) model, as it is easy to implement in simulation environments and widely used by researchers. Mobile nodes select a destination randomly and move at a speed ranging from zero to the maximum speed in meters per second towards the chosen destination. They then pause for a particular duration before continuing the same process. Before initiating our simulation, nodes' lowest and highest speeds were defined to simulate real-world scenarios. The zero mobility metric is used to deploy nodes in static scenarios, while different mobility speeds of nodes are considered for non-static scenarios. Our proposed routing technique considered both scenarios in the simulation. The mobility speeds considered are as follows: 0–1 m/s, 2–3 m/s, 4–5 m/s, and 6–10 m/s [48].

5.2. Battery Model

Battery energy models provide current and voltage to the components of wireless devices, including Radio Interfaces (RI), Central Processing Units (CPUs), and Memory Sets (MS). In addition, these models are used to predict and analyze the remaining battery capacity of nodes. The overall energy consumption per cycle is presented below:

$$\text{Cycle}_E = \text{TX}_E + \text{CPU}_E + \text{DC}_E + \text{BAT}_E \quad (1)$$

In our proposed routing scheme, a linear battery energy model is configured in each node for the estimation of residual battery energy using (1), where TX_E , CPU_E , DC_E , and BAT_E represent the energy consumed by the transceiver, central processing unit, DC converter, and battery energy efficiency loss, respectively. To account for various nodes' statuses, namely best, better, good, and bad, which return as a single value of 1, 2, 3, and 4, respectively, in OLSRV2 Hello and Topology Control (TC) messages, we configured different battery lives for nodes, as presented in Table 1.

Table 1. Simulation parameters.

Parameters	Values
Operating system	Ubuntu 22.04.2
Simulator	NS-3 Version 3.34
Simulation area	1000 × 1000 meters
Node deployment	Random positioning
No. of nodes	10, 30, 50 and 1000
Simulation time	1800 s
Mobility model	Random Waypoint Generic energy model
Energy model	Node _{TX} _CURRENT = 0.380 A/380 mA Node _{RX} _CURRENT = 0.313 A/313 Ma Node _{IDLE} _CURRENT = 0.273 A/273 mA Node _{SLEEP} _CURRENT = 0.033/33 mA
Battery model	Linear battery model
Transmission range	60m
Mac protocol	IEEE 802.11
Packet size	512
Parameters for MPR Selection [44,48]	
Battery levels	(1–24%), (25–49%), (50–74%), and (75–100%)
Node degree	(1–14%), (15–24%), (25–38%), and (39–50%)
Mobility speed	Speed: 0–1 m/s, 2–3 m/s, 4–5 m/s, and 6–10 m/s
Connection to base station	A value of 1 if a node is connected and 0 if it is not connected

5.3. Energy Consumption Model

The D2D communication networks evaluated in this paper were modeled by a graph $G(V, E)$, comprising two sets: V (nodes) and E (arcs). The arcs depict the wireless radio range between sets of wireless devices. When nodes $A1$ and $A2 \in V$ are neighboring nodes with symmetric connections with each other, then direct communication is possible. However, suppose the connection between the nodes is asymmetric or the nodes are not within the range of each other due to mobility, for example, then in that case, intermediate nodes are employed for communication using routing protocols. The network is formed through the independent broadcast of Hello and TC messages by nodes in the NS-3 simulator, as shown in Figure 13.

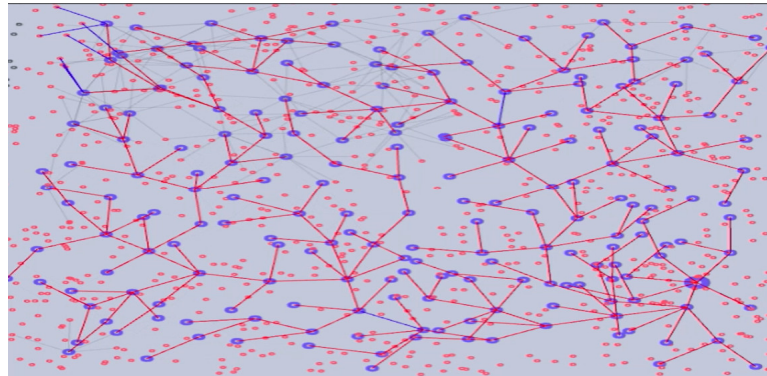


Figure 13. Deployment of nodes in NS-3 simulator via Python visualizer.

D2D communication requires devices to transmit and receive data, which can drain battery life quickly. Therefore, energy consumption is a critical concern in wireless communication, particularly for battery-powered devices in D2D networks. Energy (E) = Power \times Time (T), but P = Current (I) \times Voltage (V); therefore, $E = I V T$. However, breaking energy into transmission, receiving, idle, and sleep energy gives

$$E_T = I_T VT \tag{2}$$

$$E_R = I_R VT \tag{3}$$

$$E_D = I_D VT \tag{4}$$

$$E_S = I_S VT \tag{5}$$

where I_T , I_R , I_D , and I_S are transmission, received, idle and sleep current, respectively, while I is the sum of the current in all states. Similarly, E_T , E_R , E_D , and E_S are transmission, received, idle, and sleep energy. Therefore, to compute the total energy consumed per node while transmitting, receiving, as well as in idle and sleep modes, is carried out by adding Equations (1)–(4). The energy model considered is based on the model reported by relevant studies, including [6,7]. The total current of a node is the sum of the current while receiving, transmitting, or being in idle or sleep mode, which can be expressed as follows:

$$T_I = I_T + I_R + I_D + I_S. \tag{6}$$

The proposed energy consumption model is obtained by introducing some modifications to the OLSRv2. These modifications minimize energy consumption in the D2D communication routing protocol. Before presenting the proposed energy model, we compute the total time spent by each node transmitting or receiving by considering the known broadcast frequency of HELLO and TC messages for the proposed model, which can be expressed as follows:

$$P_1 = T_{\max}(1/NH_f + 1/H_M) + TC_m \tag{7}$$

$$P_2 = N_p T_{\max} (1/H_M + 1/NH_f) + TC_M \tag{8}$$

where P_1 is the total seconds spent transmitting data with modified algorithms, P_2 is the total seconds spent receiving data with modified algorithms, N_p is the number of Multi-Layer Point (MPR) nodes, H_M is the modified HELLO message frequency, NH_f is the new HI message frequency, TC_M is the modified TC message frequency, and T_{\max} is the total time in seconds. The proposed modification allows MPR nodes to simply extend the TTL (time to leave) of the existing TC message if no new node joins the network. OLSRv2 implementation is plagued by enormous packet loss and high processor loads due to the uncontrolled broadcasts of control messages. However, enforcing listening periods before and between message broadcasts will reduce the frequency of usage of the radio transmitter since nodes that transmit HELLO messages and fail to receive a timely response (due to

ongoing DB chores mode) will simply drop into listening mode. The reduced usage of the radio transmitter increases the energy available to the D2D nodes since transmitting consumes more energy than receiving. Therefore, using Equations (7) and (8), the new energy model that calculates the total energy consumed by nodes at a specific time is given as

$$E_{Total} = \alpha \frac{T_1 V}{N} \sum_{i=1}^3 P_i N * T_{max} \tag{9}$$

where $i = 1, 2, 3$, P_3 is the packet transmission rate, V is the node’s voltage power, α is the inefficiency factor for various nodes [25], N is the number of nodes, and T_1 is the sum of the default current [49]. The proposed message sequence is designed to decongest OLSRv2 message broadcasts and enhance information base maintenance.

5.4. Routing Overhead Model

Routing overhead in the D2D routing protocol refers to control data, messages, or operational resources necessary for the protocol to establish, share, and maintain routes in the network [50]. It constitutes additional data traffic created by the routing protocol to establish, share, and maintain routes within a network. These overheads include periodic Hello messages to establish and maintain neighbor relationships, Multiple-Interface Declaration (MID) and Host Network Association (HNA) messages, as well as the exchange of TC messages to disseminate up-to-date information about network topology [42]. The proposed routing scheme modified the sequence of OLSRv2’s HELLO, HELLO, and TC messages with a new sequence that interweaves HELLO and TC messages with three database (DB) chores and LISTEN states. This algorithm modifies the frequency of Hello and TC messages in the conventional OLSRv2, thereby changing the duration and frequency of control messages. Therefore, the new equation for a periodic control message (PCM) is given as follows:

$$PCM = \frac{F_i N}{P_3 MP_m} \tag{10}$$

where MP_m is the modified Periodic Message Interval, and F_i is the routing Control Overhead Impulse factor, and N is the number of nodes. Using Equation (10) with the modifications, the total routing overhead for the proposed routing protocol as a result of periodic control messages is given as follows:

$$PCM = \begin{cases} \frac{F_i N}{P_3 MP_m}, & \text{Hello} = 5n, MP_m = 10n, \text{ and } TC = 18n \\ 0 & \text{Otherwise} \end{cases} \tag{11}$$

where n is the number of times the messages are repeated. The modified total routing control overhead for the nodes in a network using the proposed routing protocol is presented in the equation below:

$$CO_r = P_h^l (MP_m) N p(MP_m) + \frac{\lceil \frac{U^m}{MP_m} \rceil}{\frac{U^m}{MP_m}} + \frac{F_i N}{P_3 MP_m} \tag{12}$$

where UC_t is the triggered update control message, U^m is the update message, P_m is the Periodic Message Interval, $\lceil \cdot \rceil$ is the ceiling operator, and P_h^l is the likelihood (probability) that the uplink state remains unchanged and does not switch to the downlink state during the h hops.

6. Simulation Setup

This section elucidates the implementation of the improved OLSRv2 in the well-known network simulator NS-3 and evaluates the performance of the proposed routing technique using the modified algorithms. In the proposed routing techniques, each node uses a number of models to configure and measure different parameters required for various tasks, such as sending, receiving, and relaying. Each model is responsible for extracting

real-time parameter values corresponding to its associated metric (e.g., energy, mobility, and node degree). Subsequently, these parameter values are utilized by the proposed routing protocol to identify optimal routing information and relay nodes to the target destination. A Constant Bit Rate (CBR) traffic model has been utilized for this study because it generates traffic at a constant rate, with packets sent at regular intervals and with a transmission rate of 5 packets/second, given as P_3 . The total seconds spent transmitting data with modified algorithms (P_1) and the total seconds spent receiving data with modified algorithms (P_2) were used. These transmission rates have been used throughout the four performance metrics considered in this paper. Seven simulations were conducted for each scenario with a network density of 10, 30, 50, and 100 nodes, and the statistical average was computed. Each simulation lasted for 1800 s, as given in Table 1. We provide a detailed analysis of the proposed routing protocol, focusing on four performance evaluation metrics: routing overhead, energy consumption, packet delivery, and end-to-end delay. The simulation models highlighted in this paper are essential tools for studying complex systems. These models provide a virtual environment, enabling controlled experiments and observations without the limitations of physical implementation. Our carefully designed simulation models aim to replicate real-world scenarios, contributing to a deeper understanding of dynamic interactions and intricate behaviors.

7. Result Discussion

This section presents the analysis of our simulation results and compares them with the OLSR, OLSRv2, and other improved OLSR results in the related literature, such as in [42]. The simulation parameters and their values are presented in Table 1 Section 5.2 The simulations were conducted in a static and mobile environment using network scenarios with 10, 30, 50, and 100 nodes. The results show that our improved OLSR outperforms the existing OLSR, OLSRv2, and other improved routing protocols in optimizing energy consumption, routing overhead, packet delivery ratio, and end-to-end delay.

The simulations for energy consumption in static scenarios and scenarios with various mobility speeds were carried out by implementing the modified variables and their values into the simulation environment. The simulation results for the static scenario, as shown in Figure 14, indicate that our proposed routing protocol outperforms OLSR and OLSRv2 in energy consumption across static scenarios with 10, 30, 50, and 100 nodes. This superiority is attributed to reduced overhearing, improved link management, and optimized information base management. The proposed protocol achieves this by implementing efficient neighbor discovery, minimizing irrelevant message processing, and ensuring that only active and reliable links are used. Programmable listening and base chore periods further enhance efficiency by managing information bases effectively. For scenarios with various mobility speeds, our analysis highlights that our proposed protocol effectively considers the mobility speed of nodes in the selection of the MPR, resulting in efficient energy consumption as compared to the OLSR and OLSRv2, as shown in Figure 15. By dynamically adjusting MPR selection based on the node's energy and mobility, our protocol optimizes routing decisions, reducing unnecessary overhead and conserving energy. This adaptive approach ensures efficient communication even in dynamic environments, enhancing network performance and longevity. In general, the energy consumption of the proposed routing protocol is quite impressive, taking into account the huge packet delivery in all scenarios and, therefore, demonstrating robustness and energy efficiency as compared to the energy consumed by nodes in related research such as the work of [42].

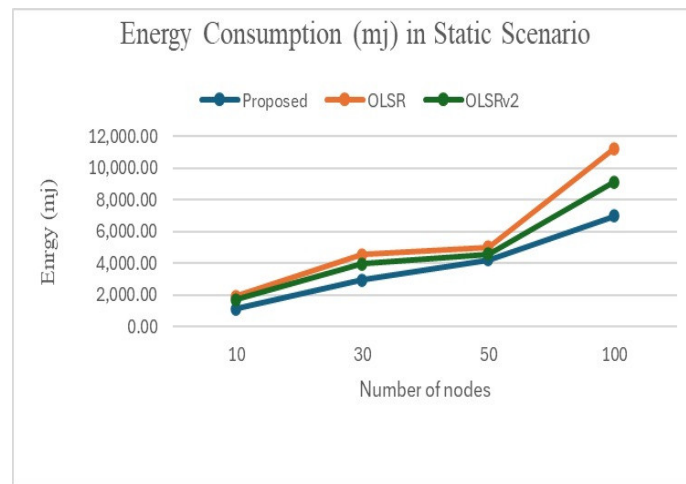


Figure 14. Energy consumption against number of nodes in static scenario.

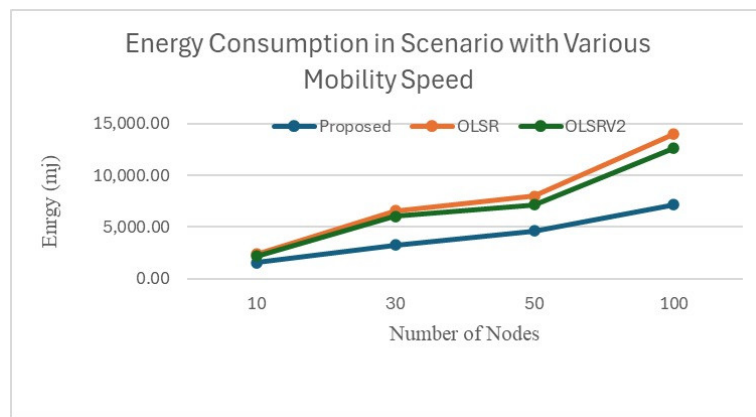


Figure 15. Energy consumption against number of nodes in mobility scenario.

The simulation for routing overhead was also conducted for both static scenarios and scenarios with various mobility speeds for scenarios with 10, 30, 50, and 100 nodes. The associated network models for periodic control messages and the total routing control overhead are given in Equations (11) and (12), respectively. The results of our simulation for routing overhead in static scenarios are presented in Figure 16 and they show that our proposed routing protocol significantly outperforms OLSR and OLSRv2 in terms of routing overhead. This superiority is attributed to optimized message exchange strategies, efficient neighbor discovery mechanisms, and dynamic multipoint relay (MPR) selection based on node status. These factors reduce the number of control messages and unnecessary message propagation, resulting in lower routing overhead. Similarly, in our simulation with mobility speeds ranging from 0 to 1 m/s, 2 to 3 m/s, 4 to 5 m/s, and 6 to 10 m/s, our proposed protocol consistently demonstrates lower routing overhead compared to OLSR, OLSRv2, and other improved OLSR versions such as in [42], as displayed in Figure 17. Traditional OLSR and OLSRv2 experience significantly higher routing overhead due to node movements and topology changes. In contrast, our protocol maintains efficient routing with minimal overhead across all network densities, highlighting its robustness and suitability for dynamic wireless environments, thereby offering a more streamlined routing process, leading to improved network scalability and resource efficiency compared to the standard protocols.

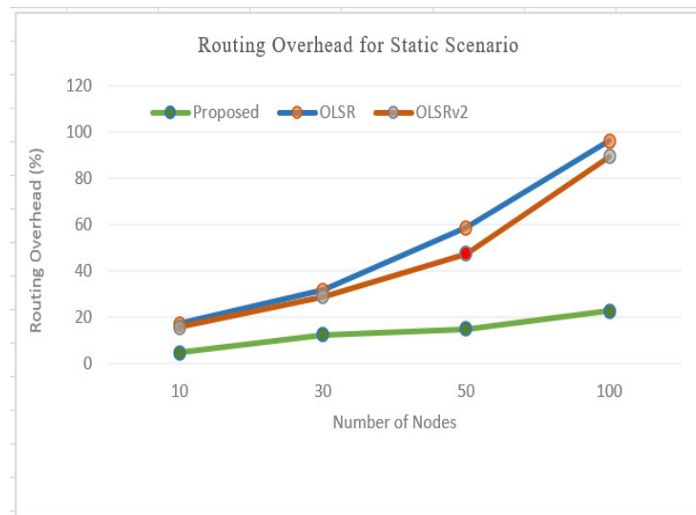


Figure 16. Routing overhead against number of nodes in static scenario.

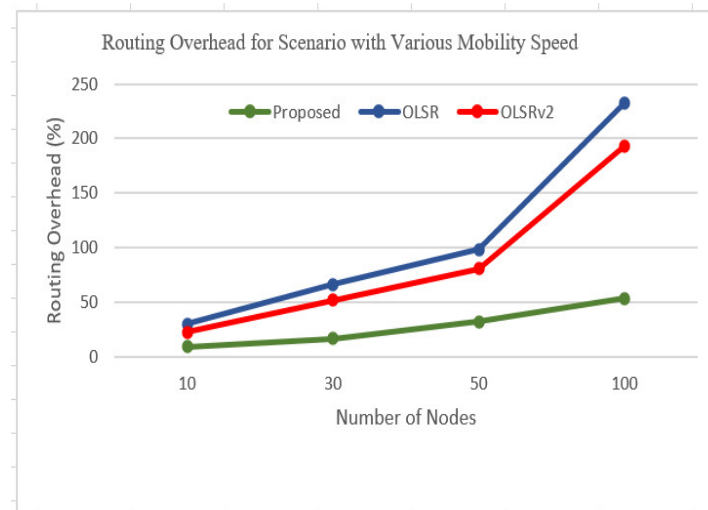


Figure 17. Routing overhead against number of nodes in mobility scenario.

Figure 18 reveals that our proposed routing protocol achieves a higher packet delivery ratio (PDR) compared to OLSR and OLSRv2, attributed to reduced overhearing, improved link management, and proper information base management. By implementing efficient neighbor discovery mechanisms and optimized message exchange strategies, our protocol reduces interference and improves transmission reliability, leading to a higher PDR. Additionally, the protocol focuses on using only reliable and stable links for communication, further boosting the PDR. The programmable listen and base chore periods before and after message broadcasts ensure efficient information base management, resulting in better routing decisions and fewer packet losses. These improvements highlight the protocol’s effectiveness in enhancing packet delivery and network performance compared to OLSR and OLSRv2. The simulation results indicate that our proposed protocol outperforms the classical versions of OLSR and the latest improved version of OLSR given in [42] in terms of packet delivery ratio (PDR) across various mobility speeds, as shown in Figure 19. OLSR and OLSRv2 experience a decrease in PDR due to node movements and topology changes. In contrast, our protocol maintains a higher PDR, demonstrating its effectiveness in handling dynamic wireless environments and ensuring reliable packet delivery compared to the standard protocols.

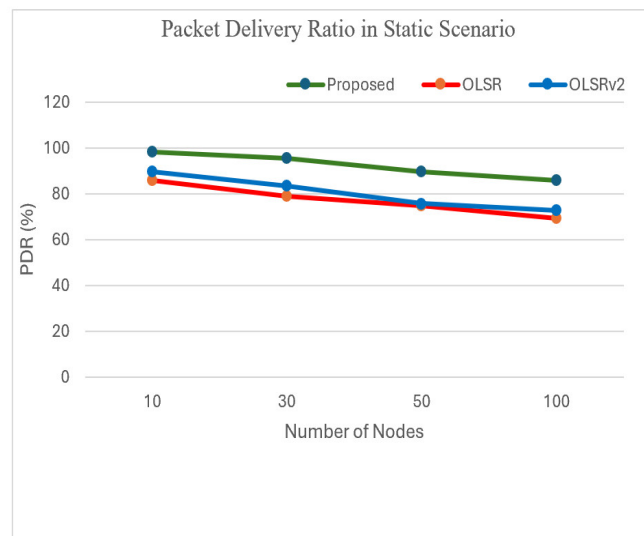


Figure 18. Packet delivery ratio against number of nodes in mobility scenario.

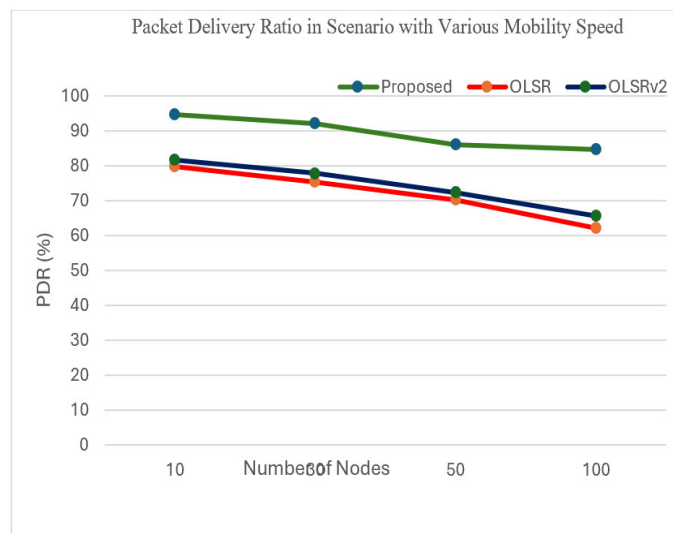


Figure 19. Packet delivery ratio against number of nodes in static scenario.

Our proposed protocol exhibits superior performance in end-to-end delay compared to the classical versions of OLSR and the latest improved version of OLSR given in [42], primarily due to reduced overhearing, improved link management, and efficient information base management, as in Figure 20. By implementing optimized neighbor discovery mechanisms and message exchange strategies, our protocol minimizes unnecessary message propagation and interference, leading to shorter end-to-end delay. Additionally, the protocol’s focus on utilizing only active and reliable links for communication enhances link stability, further reducing delay. The use of programmable listen and base chore periods before and after message broadcasts ensures effective information base management, contributing to quicker routing decisions and reduced delay. Overall, these enhancements result in our protocol performing better in end-to-end delay compared to OLSR and OLSRv2 in static scenarios. With the implementation of mobility speeds, our proposed routing protocol demonstrates superior performance in end-to-end delay compared to OLSR, OLSRv2, and the work of [42] across various mobility speeds, as in Figure 21. OLSR and OLSRv2 experience higher delays due to increased node movements and topology changes. In contrast, our protocol maintains lower end-to-end delay across all scenarios, showcasing its effectiveness in handling dynamic wireless environments and ensuring quicker data packet delivery compared to the conventional protocols.

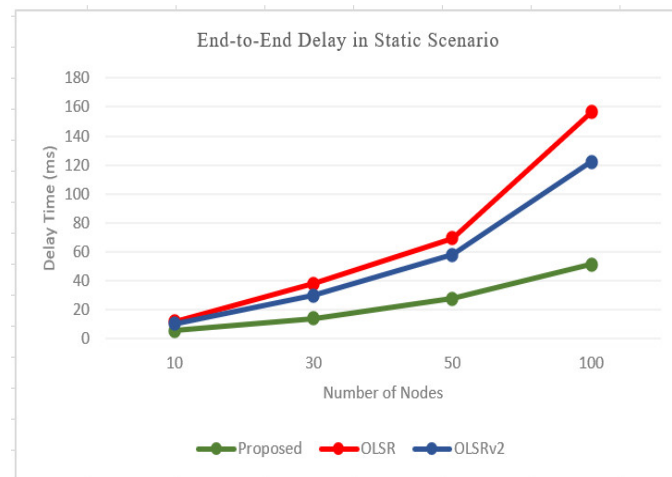


Figure 20. End-to-end delay against number of nodes in static scenario.

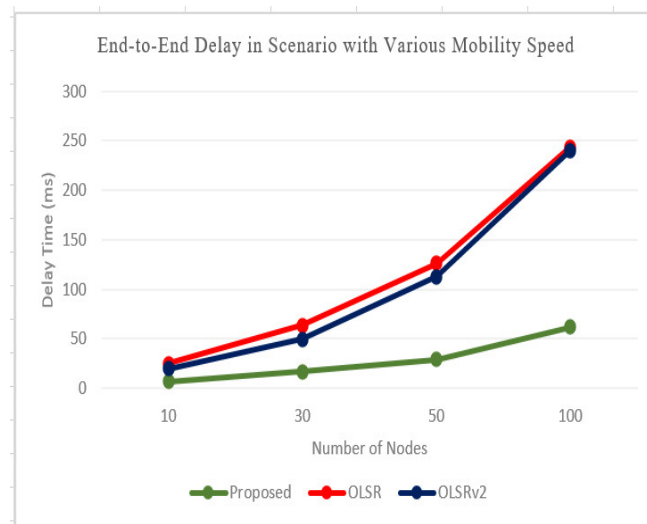


Figure 21. End-to-end delay against number of nodes in various mobility speeds.

8. Conclusions

In conclusion, our proposed routing protocol demonstrates significant improvements in energy efficiency, routing overhead, packet delivery ratio (PDR), and end-to-end delay compared to the standard OLSR and OLSRv2 protocols and other improved OLSR results given in the related literature, such as in the work of [42]. The protocol's efficiency is attributed to its ability to reduce overhearing, enhance link management, and optimize information base management through programmable listening and base chore periods. These enhancements make our protocol well-suited for dynamic wireless environments, ensuring reliable and efficient communication. Overall, our protocol offers a promising solution for improving the performance of D2D communication networks, paving the way for enhanced network scalability and reliability. For future work, several areas could be explored to further enhance the performance and capabilities of our proposed routing protocol. Firstly, we will investigate the protocol's performance in larger-scale networks with hundreds or thousands of nodes. This could provide valuable insights into its capacity, scalability, and robustness. Additionally, integrating machine learning techniques for dynamic MPR selection and routing decisions could further optimize the protocol's performance, especially in highly dynamic environments. Furthermore, conducting real-world experiments or field trials to validate the protocol's performance under realistic conditions would be beneficial.

Author Contributions: Conceptualization, S.M.B.; methodology, S.M.B.; software, S.M.B.; validation, S.M.B.; resources, S.M.B.; data curation, S.M.B.; writing original draft preparation, S.M.B. Formal analysis, O.Y.A. and M.S.; Investigation, O.Y.A. and M.S.; Review and Editing, O.Y.A. and M.S.; Supervision, O.Y.A. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Ron, D.; Lee, J.-R. Learning-based joint optimization of mode selection and transmit power control for D2D communication underlaid cellular networks. *Expert Syst. Appl.* **2022**, *198*, 116725. [CrossRef]
- Usman, M.; Gebremariam, A.A.; Raza, U.; Granelli, F. A software-defined device-to-device communication architecture for public safety applications in 5G networks. *IEEE Access* **2015**, *3*, 1649–1654. [CrossRef]
- Doumiati, S.; Artail, H.; Gutierrez-Estevez, D.M. A framework for LTE-A proximity-based device-to-device service registration and discovery. *Procedia Comput. Sci.* **2014**, *34*, 87–94. [CrossRef]
- Piro, G.; Orsino, A.; Campolo, C.; Araniti, G.; Boggia, G.; Molinaro, A. D2D in LTE vehicular networking: System model and upper bound performance. In Proceedings of the 2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Brno, Czech Republic, 6–8 October 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 281–286.
- Wen, S.; Zhu, X.; Lin, Y.; Lin, Z.; Zhang, X.; Yang, D. Achievable transmission capacity of relay-assisted device-to-device (D2D) communication underlay cellular networks. In Proceedings of the 2013 IEEE 78th Vehicular Technology Conference (VTC Fall), Las Vegas, NV, USA, 2–5 September 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–5.
- Al-Kanj, L.; Poor, H.V.; Dawy, Z. Optimal cellular offloading via device-to-device communication networks with fairness constraints. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 4628–4643. [CrossRef]
- Malathy, S.; Jayarajan, P.; Hindia, M.N.; Tilwari, V.; Dimiyati, K.; Noordin, K.A.; Amiri, I.S. Routing constraints in the device-to-device communication for beyond IoT 5G networks: A review. *Wirel. Netw.* **2021**, *27*, 3207–3231. [CrossRef]
- Pearlman, M.R.; Haas, Z.J. Determining the optimal configuration for the zone routing protocol. *IEEE J. Sel. Areas Commun.* **1999**, *17*, 1395–1414. [CrossRef]
- Tilwari, V.; Dimiyati, K.; Hindia, M.N.; Fattouh, A.; Amiri, I.S. Mobility, residual energy, and link quality aware multipath routing in MANETs with Q-learning algorithm. *Appl. Sci.* **2019**, *9*, 1582. [CrossRef]
- Ghahfarokhi, B.S.; Azadmanesh, M.; Khorasani, S.K. Energy and spectrum efficient mobility-aware resource management for D2D multicasting. *Comput. Netw.* **2018**, *146*, 47–64. [CrossRef]
- Guo, H.; Zhou, X.; Liu, J.; Zhang, Y. Vehicular intelligence in 6G: Networking, communications, and computing. *Veh. Commun.* **2022**, *33*, 100399. [CrossRef]
- Shukla, S.; Bhardwaj, O.; Abouzeid, A.A.; Salonidis, T.; He, T. Proactive retention-aware caching with multi-path routing for wireless edge networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1286–1299. [CrossRef]
- Siraj, M.N.; Ahmed, Z.; Hanif, M.K.; Chaudary, M.H.; Khan, S.A.; Javaid, N. A hybrid routing protocol for wireless distributed networks. *IEEE Access* **2018**, *6*, 67244–67260. [CrossRef]
- Ahmed, R.E. A Novel Multi-Hop Routing Protocol for D2D Communications in 5G. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 27–30 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 0627–0630.
- Clausen, T.; Jacquet, P. RFC3626: *Optimized Link State Routing Protocol (OLSR)*; Experimental; RFC Editor: Marina del Rey, CA, USA, 2003; Volume 51, Available online: <https://www.ietf.org/rfc/rfc3626.txt> (accessed on 10 June 2024).
- He, G. *Destination-Sequenced Distance Vector (DSDV) Protocol*; Networking Laboratory, Helsinki University of Technology: Espoo, Finland, 2022; pp. 1–9.
- Govindasamy, J.; Punniakody, S. A comparative study of reactive, proactive and hybrid routing protocol in wireless sensor network under wormhole attack. *J. Electr. Syst. Inf. Technol.* **2018**, *5*, 735–744. [CrossRef]
- Taha, A.; Alsaqour, R.; Uddin, M.; Abdelhaq, M.; Saba, T. Energy efficient multipath routing protocol for mobile ad-hoc network using the fitness function. *IEEE Access* **2017**, *5*, 10369–10381. [CrossRef]
- Das, S.R.; Belding-Royer, E.M.; Perkins, C.E. *Ad Hoc On-Demand Distance Vector (AODV) Routing*; RFC Editor: Marina del Rey, CA, USA, 2003.
- Johnson, D.; Hu, Y.-C.; Maltz, D. *The Dynamic Source Routing Protocol (DSR) for Mobile ad Hoc Networks for IPv4*; 2070-1721; RFC Editor: Marina del Rey, CA, USA, 2007.
- Pandey, A.; Ahmed, M.N.; Kumar, N.; Gupta, P. A hybrid routing scheme for mobile ad hoc networks with mobile backbones. In Proceedings of the International Conference on High-Performance Computing, Bangalore, India, 18–21 December 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 411–423.
- Ali, M.; Qaisar, S.; Naem, M.; Mumtaz, S.; Rodrigues, J.J. Combinatorial resource allocation in D2D assisted heterogeneous relay networks. *Future Gener. Comput. Syst.* **2020**, *107*, 956–964. [CrossRef]

23. Wheeb, A.H.; Nordin, R.; Samah, A.A.; Kanellopoulos, D. Performance Evaluation of Standard and Modified OLSR Protocols for Uncoordinated UAV Ad-Hoc Networks in Search and Rescue Environments. *Electronics* **2023**, *12*, 1334. [[CrossRef](#)]
24. Aliyu, U.; Takruri, H.; Hope, M.; Halilu, A.G. DS-OLSR–Disaster Scenario Optimized Link State Routing Protocol. In Proceedings of the 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Porto, Portugal, 20–22 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
25. Jabbar, W.A.; Saad, W.K.; Ismail, M. MEQSA-OLSRv2: A Multicriteria-Based Hybrid Multipath Protocol for Energy-Efficient and QoS-Aware Data Routing in MANET-WSN Convergence Scenarios of IoT. *IEEE Access* **2018**, *6*, 76546–76572. [[CrossRef](#)]
26. Mitra, R.; Sharma, S. Proactive data routing using controlled mobility of a mobile sink in wireless sensor networks. *Comput. Electr. Eng.* **2018**, *70*, 21–36. [[CrossRef](#)]
27. Zeng, Y.; Zhou, J.; Liu, Y.; Cao, T.; Yang, D.; Liu, Y.; Shi, X. Stable routing protocol for unmanned aerial vehicle ad-hoc networks based on DQN-OLSR. *IET Commun.* **2023**, *17*, 73–85. [[CrossRef](#)]
28. Elektra. The OLSR.ORG Story. Available online: <https://www.open-mesh.org/projects/open-mesh/wiki/The-olsr-story> (accessed on 6 May 2023).
29. Huang, Y.; Xie, R.; Gao, B.; Wang, J. Dynamic Routing in Flying Ad-hoc Networks using Link Duration Based MPR Selection. In Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Virtual, 18 November–16 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
30. Cadeiro, W. OLSR (RFC 3626 + ETX / ML / MD Variants) for NS-2. Available online: <https://github.com/wevertoncordeiro/olsr-ns2> (accessed on 5 September 2023).
31. Aliyu, U. Disaster Scenario Optimised Link State Routing Protocol for Disaster Recovery and Rescue Operations. Ph.D. Thesis, University of Salford, Salford, UK, 2021.
32. Kazmi, S.H.A.; Qamar, F.; Hassan, R.; Nisar, K. Routing-based interference mitigation in SDN enabled beyond 5G communication networks: A comprehensive survey. *IEEE Access* **2023**, *11*, 4023–4041. [[CrossRef](#)]
33. Gustavo, C. Optimized Link State Routing (OLSR). Available online: <https://www.nsnam.org/docs/models/html/olsr.html> (accessed on 7 April 2023).
34. Francisco, R. UM-OLSR, an Implementation of the OLSR Routing Protocol for the NS2 Network Simulator. Available online: <https://www.nsnam.org/docs/release/3.32/models/html/olsr.html> (accessed on 7 April 2023).
35. Ros, F.J.; Ruiz, P.M. *Implementing a New Manet Unicast Routing Protocol in NS2*; Department of Information and Communications, Engineering University of Murcia: Murcia, Spain, 2004; Volume 35.
36. Hamzaoui, Y.; Chekour, M.; Amnai, M. Enhanced MPR Selection Algorithm Based on Metrics for the OLSR Routing Protocol. In Proceedings of the 2023 6th International Conference on Advanced Communication Technologies and Networking (CommNet), Rabat, Morocco, 11–13 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
37. Hamzaoui, Y.; Amnai, M.; Choukri, A.; Fakhri, Y. Enhancenig OLSR routing protocol using K-means clustering in MANETs. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 3715. [[CrossRef](#)]
38. Kumari, P.; Sahana, S.K. Swarm based hybrid ACO-PSO meta-heuristic (HAPM) for QoS multicast routing optimization in MANETs. *Wirel. Pers. Commun.* **2022**, *123*, 1145–1167. [[CrossRef](#)]
39. Oyakhire, O.; Gyoda, K. Improved proactive routing protocol considering node density using game theory in dense networks. *Future Internet* **2020**, *12*, 47. [[CrossRef](#)]
40. Zhang, C.; Wu, C.; Lin, M.; Lin, Y.; Liu, W. Proximal Policy Optimization for Efficient D2D-Assisted Computation Offloading and Resource Allocation in Multi-Access Edge Computing. *Future Internet* **2024**, *16*, 19. [[CrossRef](#)]
41. Maccari, L.; Cigno, R.L. How to reduce and stabilize MPR sets in OLSR networks. In Proceedings of the 2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, 8–10 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 373–380.
42. Aliyu, U.; Takruri, H.; Hope, M.; Gidado, A.H.; Adamu, H.A. Disaster scenario optimised link state routing protocol and message prioritisation. *IET Netw.* **2024**. Available online: <https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/ntw2.12125> (accessed on 9 September 2024).
43. Rahul, P.; Kaarthick, B. Proficient link state routing in Mobile ad hoc network-based Deep Q-learning network optimized with chaotic bat swarm optimization algorithm. *Int. J. Commun. Syst.* **2023**, *36*, e5324. [[CrossRef](#)]
44. Bunu, S.; Saraee, M.; Alani, O. Optimizing the Parameters of Relay Selection Model in OLSRv2 D2D Network. In Proceedings of the 2023 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Batam, Indonesia, 11 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 457–462.
45. Clausen, T.; Dearlove, C.; Jacquet, P.; Herberg, U. *The Optimized Link State Routing Protocol Version 2*; 2070-1721; RFC Editor: Marina del Rey, CA, USA, 2014.
46. Clausen, T.; Dearlove, C.; Dean, J. *Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)*; 2070-1721; RFC Editor: Marina del Rey, CA, USA, 2011.
47. Clausen, T.; Dearlove, C.; Dean, J.; Adjih, C. *Generalized Mobile ad Hoc Network (MANET) Packet/Message Format*; 2070-1721; RFC Editor: Marina del Rey, CA, USA, 2009.
48. Bunu, M.; Saraee, M.; Alani, O. Machine Learning-Based Optimized Link State Routing Protocol for D2D Communication in 5G/B5G. In Proceedings of the 2022 International Conference on Electrical Engineering and Informatics (ICELTICs), Banda Aceh, Indonesia, 27–28 September 2022; pp. 7–12. [[CrossRef](#)]

-
49. Bunu, S.M.; Saraee, M.; Alani, O. A novel mathematical model for optimizing energy consumption in OLSRv2 routing protocol. In Proceedings of the 2024 4th International Conference of Science and Information Technology in Smart Administration (ICSINTESA), Balikpapan, Indonesia, 12–13 July 2024. *in press*.
 50. Al-Zahrani, F.A. On modeling optimizations and enhancing routing protocols for wireless multihop networks. *IEEE Access* **2020**, *8*, 68953–68973. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.