Received: January 4, 2024. Revised: February 1, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Efficient OLSR Routing Protocol to Minimize Multipoint Relays in MANET

Md. Zahid Hassan[1]* **Shahid Md. Asif Iqbal**[2] **Asaduzzaman**[1]

[1]*Chittagong University of Engineering and Technology, Bangladesh*
[2]*Premier University, Bangladesh*
* Corresponding author's Email: zahidhassan956@gmail.com

Abstract: Reducing control packets, especially in proactive routing protocols, needed to establish routes can lower network overhead in Mobile Ad-hoc Networks (*MANETs*). In Optimized Link State Routing (*OLSR*) protocol, each Multipoint Relay (*MPR*) node propagates Topology Control (*TC*) messages to advertise neighbor information. However, *OLSR* controls the *TC* messages by reducing the number of *MPR* nodes. In this study, we propose an efficient *MPR* node selection mechanism to reduce the *TC* message volume leading to a minimized routing overhead. Each node selects the lowest cost node heuristically from its first hop neighbors as the *MPR* node for any destination. The same *MPR* node can be selected for multiple destinations if it costs the lowest for each destination node. The selection technique is realized by modifying only the default *OLSR TC* and *Hello* messages. The proof-of-concept implementation in the *NS3* simulator reveals that the proposed methodology reduces the routing overhead by selecting around 55%, 28% and 49% (on average) fewer *MPR* nodes compared to the traditional *OLSR*, *SSTB* and *M-OLSR* protocol respectively, without negotiating packet delivery ratio, throughput and delay.

Keywords: *MANET*, *OLSR*, *TC* Messages, Routing overhead, *MPR*.

1. Introduction

MANET [1] is a variant of ad-hoc networks where nodes are mobile and decentralized in type, and packet routing does not need any pre-established centralized infrastructure. *MANET* is autonomous, self-configurable, and highly adaptive, and the distinct features make it ideal for realization in scenarios where an infrastructure network is absent or failed, or establishment is challenging or impossible, for example, military applications [2], forest fire surveillance [3], search and rescue operations [4], disaster recovery and rescue operations [5], etc. The routing protocols [6] are responsible for delivering packets and maintaining the paths between communicating nodes in *MANET* [7-9].

Routing protocols in *MANET* can broadly be categorized into proactive, reactive, and hybrid routing protocols [10]. *OLSR* [11] is one of the most popular wireless routing protocols exhibiting comparatively better performance in *MANET*, and the classical link state routing mechanism is optimized to develop *OLSR*. Being a proactive protocol, *OLSR* guarantees prior route availability every time. The prior route availability enables it to outperform its counterpart benchmarks in terms of packet delivery ratio (*PDR*), throughput, and end-to-end delay [12-15]. However, the table-driven characteristics cause *OLSR* to experience a higher routing overhead than those counterparts. Thus, the performance enhancement *OLSR* has become a highly debated research topic. This research chooses to address and improve the *OLSR* routing overhead issue without sacrificing other performance issues, for instance, *PDR*, throughput, and delay.

Nodes in *MANET* can establish and maintain required routes through a regular or periodic exchange of *Hello* and *TC* messages. However, the rise in *TC* messages, especially in dense networks, could lead to message collisions, traffic congestion, and increased energy use, which are potential reasons for performance degradation. *OLSR* controls or optimizes the *TC* message broadcasting by permitting only the selected *MPR* nodes to forward *TC* messages.

A single *TC* packet dispensed by an *MPR* node may encapsulate two or more *TC* messages, which aids in lowering the routing overhead and the likelihood of packet collision from different nodes. Thus, reducing the *MPR* set can reduce the number of *TC* messages.

The traditional *MPR* selection algorithm is unsuitable for keeping the *MPR* set small as it selects more *MPR* nodes needed to cover all possible 2-hop neighbors. A few heuristic solutions for selecting the best *MPR* are proposed in the literature; however, the schemes are sophisticated, challenging to use, and consume additional resources. Therefore, this work proposes an improved *MPR* selection technique covering only one-hop neighbors and effectively decreasing the number of control packets without sacrificing other performance metrics and is implemented by a network simulator named *NS3* (*NS-3.30*) [16]. The key contributions of this paper can be summarized as follows:

The size of *MPR* set is reduced since only the lowest cost node/s in the first-hop neighbor is considered as the *MPR* node/s.

The same *MPR* node can be used for multiple destinations if it is the lowest-cost node for each destination.

Only the default control messages are extended to realize the proposed strategy.

The proposed strategy is contrasted against the default *OLSR* and *M-OLSR* in terms of *PDR*, throughput, delay, and overhead, by varying the number of nodes and pause time.

The remaining part of this paper is organized as follows. Section 2 reviews the related literature of different optimizations in existing *OLSR*. The system model, assumptions, and problem formulation have been discussed in section 3. Section 4 presents the working methodology. Section 5 demonstrates the simulation results and finally, section 6 concludes the paper.

2. Related work

Several different sorts of research have been done in the last few decades to enhance the *OLSR* protocol's functionality on *MANET* networks. For enhancing the performance, researchers have focused more attentions in *MPR* selection strategy to reduce routing overhead in the network. Being a proactive protocol, *OLSR* maintains route quality and experience lower latency than their reactive counterparts, such as *DSR* and *AODV*, as routing information is available anytime. However, the proactive protocols show deteriorated performance regarding routing overhead [17]. This section explores past efforts that made similar contributions to several *OLSR* routing schemes in ad-hoc networks.

The authors of this paper [18], introduce a new process of choosing *MPR* nodes, named *M-OLSR*, by giving higher priority to nodes that are more stable in terms of energy and mobility. The objective of this approach is to improve overall network performance by incorporating a mobility metric into the traditional *MPR* selection procedure. Based on the mobility degree captured or the node with the largest residual energy, this protocol gives priority to less mobile candidate *MPR* nodes. The drawback of this strategy is that, depending on the flow of motion around the node, the parameter λ (coefficient of flow) must be fixed between three values (0.25, 0.5, and 0.75). *M-OLSR* does not, however, adequately reduce the routing overhead (Fig. 12).

In [19], a new strategy called "Selector Set Tie Breaker" (*SSTB*) has been proposed for minimizing the global *MPR* set (the union of all the *MPR* sets). Prior to implementing the initial tie-break [20], an additional step is included that essentially favours the node with the greatest number of selectors among *MPR*s and the node that is already an *MPR* for another node. However, this mechanism reduces fewer number of *MPR* set compared to original *OLSR*, without considering other performance metrics and this *MPR* selection has been optimized in this study (Fig. 11) using heuristic concepts.

In this paper [21], the authors propose a quantum-genetic-based modified *OLSR* protocol to reduce the redundant information in *MANET*. According to an improved version of the quantum genetic algorithm, they introduced a new *MPR* selection scheme in which a newly designed Q-Learning technique has been adopted, and nodes are encoded by the quantum gene bit. A heuristic node fitness rule has been followed to select a small *MPR* set for each node. In this paper, network control overhead drastically increases with network size.

AOLSR, explained in [22], offers greater *MPR* selection criteria optimization. Less overhead is accomplished by placing the *MPR* node on either the left or right side of the sender node, depending on where the destination node is located. This protocol works well in terms of packet delivery ratio and throughput.

In [23], the authors propose a swarm-based hybrid *ACO PSO* meta-heuristic (*HAPM*) routing protocol to ensure routing in large and dynamic ad hoc network. To increase *QoS* restrictions and reduce *QoS* data dropping, this protocol combines *ACO*, *PSO*, and a dynamic queue mechanism. Although this protocol works well in large scale dynamic

Received: January 4, 2024. Revised: February 1, 2024.

environment, routing overhead has not been reduced up to the mark.

Additionally, the researchers have made several excellent attempts to select *MPR*s in order to improve the performance of the *OLSR* protocol while taking packet delivery ratio, routing overhead, throughput [24-26], energy efficiency [27], security issues [28] etc. into account.

The majority of previously referenced works for improving the earlier *MPR* selection strategy defined in standard *OLSR* protocol, which increases the number of chosen *MPR* nodes as well as introduce more complexities. So, we have applied heuristic concepts in *MPR* selection process which has been able to choose less number of *MPR* nodes as well as less *TC* message propagation compared to standard *OLSR* without degrading other performances.

3. System model, assumptions, and problem formulation

This section commences by briefly picturing the working procedure of the classical *OLSR* algorithm.

3.1 Network topology

OLSR enables proactive routing to determine the best path by spreading various types of control messages such as Hello, TC, MID, and HNA. The MANET nodes exchange neighbor and routing information through the control messages. The nodes utilize the control packets to build and keep the topology information in their routing tables. The network topology in Fig. 1 illustrates the proposed MPR selection technique where data from a sender finds the best paths to the given destinations.

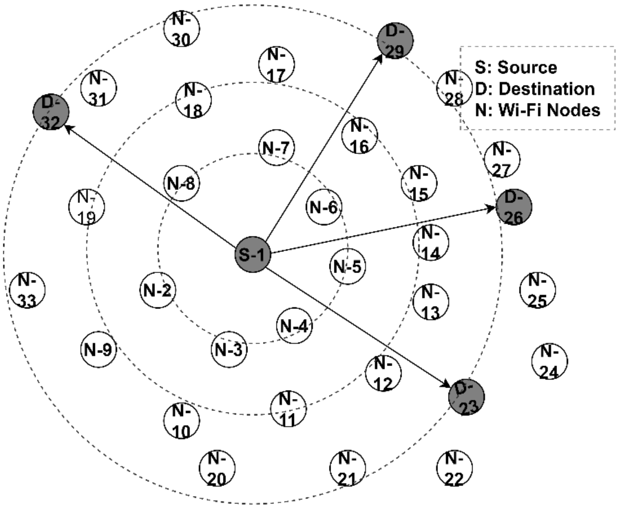

Figure. 1 Network topology

3.2 Existing *MPR* selection strategy used by *OLSR*

(*MPR*s) nodes are vital to reduce the dissemination of control *TC* messages. The classical *MPR* selection algorithm is heuristic in manner [11] where a node (u) needs to maintain its one-hop and two-hop neighbor sets, denoted as *N(u)* and *N2(u)*, respectively. *N2(u)* includes nodes reachable by the members of one-hop neighbors *N(u)*, and whose willingness is not *WILLNEVER*. Each node maintains the "willingness" parameter, an integer value that ranges from 0 to 7, indicating its eagerness to forward traffic on behalf of other nodes. Any node not interested in forwarding traffic for other nodes, such as because of resource limitations, is indicated by *WILLNEVER(0)*. *WILLALWAYS(7)* denotes that a node is always ready to carry traffic on behalf of other nodes. By default, every node has the willingness set to *WILLDEFAULT(3)*. When any node y is a member of *N(u)*, its degree is denoted as *D(y)*. *D(y)* defines the number of symmetric neighbors of node y, omitting any other nodes that are also members of *N(u)*, and the node u doing the computation. The detailed classical *MPR* selection algorithm has been given in Algorithm 1.

Algorithm 1: Classical *MPR* selection strategy defined in *RFC 3626* [11]

```
Start with $MPR(u) \leftarrow N(u)$ where willingness of $y \in N(u)$ is $WILLALWAYS$
Compute $D(y)$ for all $y \in N(u)$
for Each $y \in N(u)$ do
    if $y$ is the only node to reach some $w \in N2(u)$ then
        Add $y$ to $MPR(u)$ and Remove $w$ from $N2(u)$
    end if
end for
while $N2(u)$ remains not empty do
    if Only $y \in N(u)$ has highest reachability and willingness for some $w \in N2(u)$ then
        Add $y$ to $MPR(u)$ and Remove $w$ from $N2(u)$
        if More $y \in N(u)$ with same reachability and willingness then
            Find $y \in N(u)$ where $D(y)$ is maximum
            Add $y$ to $MPR(u)$ and Remove $w$ from $N2(u)$
        end if
    end if
end while
Integrate $MPR(u)$ for all interfaces of $u$
```

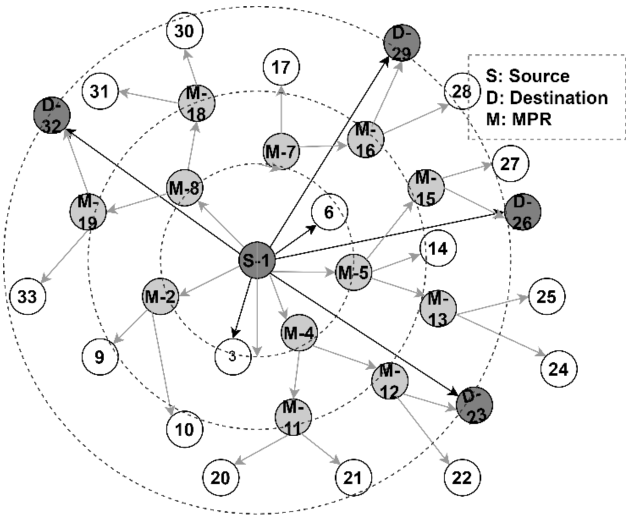

Figure. 2 Existing *OLSR*

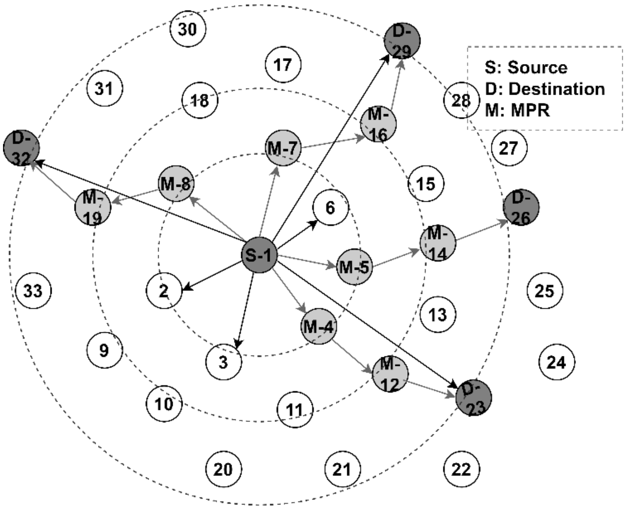

Figure. 3 Proposed *OLSR*

3.3 Problem definition

The classical *MPR* selection algorithm explained in section 3.2. results in many *MPR* nodes being selected for *TC* dissemination. Here, all the two-hop neighbors need to be covered by *MPR* nodes. However, the proposed methodology selects only those nodes as *MPR* needed to obtain optimal paths toward the destinations. The *MPR* and route selection scenarios of the classical and proposed algorithms are pictorially presented in Figs. 2 and 3 respectively. A node y in the proposed technique uses a heuristic function to select *MPR* nodes from its one-hop neighbor set, *N(u)*, explained in 4. Each node selects the lowest cost node from its *N(u)* neighbor set as the *MPR* node for a particular destination node. The same *MPR* node can be selected for multiple destinations if it costs the lowest for each destination node. Only nodes that reside along the optimal path are selected as *MPR* nodes in this process. Therefore, the number of *MPR* nodes can be drastically reduced by pruning unnecessary or sub-optimal paths toward the destinations. If n and $|MPR(y)|$ represent the number of sinks and *MPR* nodes of y, respectively, then $|MPR(y)| \leq n$ for each node, y. In contrast, in classical *OLSR*, $|MPR(y)| \propto N2(u)$. Thus, the number of *MPR* nodes selected in the proposed strategy is not dependent on the *N2(u)* set, rather it leans on the number of sinks resulting in a smaller-sized *MPR* set.

4. Proposed method

This section introduces the needed modifications of *Hello* and *TC* messages to execute the proposed technique. The modifications and the *MPR* selection strategy collectively aid in lowering the number of *MPR* nodes to diffuse fewer *TC* messages.

4.1 Extended *Hello* message format

As nodes' locations are at the heart of the proposed *MPR* selection process, every node must know its neighbors' and destination nodes' locations. In this study, each node is assumed to equipped with a *GPS* receiver to obtain its location information; longitude and latitude positions. A node maintains and shares its neighbors' and destination locations by broadcasting periodic *Hello* messages. A new table, named *Dest_Table* (Fig. 6), is introduced to maintain the destinations' location information. In addition, the default neighbor table (Fig. 4) is extended by adding two fields to store neighbors' location and node costs. Fig. 5 exhibits the proposed *Hello* message to accommodate the location information.

Location(X) and *Location(Y)* represent the *longitude(X)* and *latitude(Y)* co-ordinates, respectively of the sender node. A node retrieves its neighbors' location information once a *Hello* is received. The *NodeCost* field is used to share the link cost established for each neighbor node. *Node Cost* is calculated using Eq. (3) as explained in section 4.3.3. *IsDest* represents a boolean value that determines whether the *Hello* message's sender is a destination. *DestMsgSize* contains the size of *Dest_Table* of the sender node. This field helps a receiver node to store sender's *Dest_Table* related information. The information of each tuple in *Dest_Table* is shared through *DestinationLocation(X)*, *DestinationLocation(Y)*, and *DestinationInterfaceAddress* fields, respectively. the rest of the fields are similar to the original *Hello* message format.

Neighbor Main Address (32 bit)	Status (2 bit)	Willingness (8 bit)	Node Cost (16 bit)	Neighbor Location (32 bit)

Figure. 4 Extended *Neighbor_Table* format

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Reserved		Htime	Willingness
Location(X)		Location(Y)	
Node Cost		Is Dest	Dest Msg Size
Destination Location(X)		Destination Location(Y)	
Destination Interface Address			
. . .			
Destination Location(X)		Destination Location(Y)	
Destination Interface Address			
. . .			
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			
. . .			
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			

. . .
etc.

Figure. 5 Extended *Hello* message format

4.2 Proposed table formats

The *MPR* selection technique is realized by each node maintaining three new tables named *Dest_Table*, *MPR_Table*, and *Cost_Table*. The tables' purposes are described in the following sections.

4.2.1 Proposed table formats

Dest_Table stores information related to the specified destinations, as represented in Fig. 6. A *Hello* message uses the table's information to broadcast destination-related information. Also, the table is used for *MPR* calculation. *IPv4* address and location collected via the exchanges of *Hello* messages. When a node receives a *Hello* message, it first determines whether the sender is a destination node by inspecting the *IsDest* field of the *Hello* message. If the sender is the destination node, it updates its *Dest_Table* with the destination address and location. The node later shares the destination information by broadcasting *Hello* messages to its neighbors. The process continues, and each node is informed about the destinations once the network converges.

Destination Node Address (32 bit)	Destination Node Location (32 bit)

Figure. 6 *Dest_Table* format

MPR Selector Address (32 bit)	Destination Address (32 bit)	Cost (8 bit)	Node Cost (From Source) (8 bit)	Destination Location (32 bit)

Figure. 7 *MPR_Table* format

4.2.2 *MPR_Table* format

This table consists of five fields as represented in Fig. 7. A node's *MPRSelectorAddress* field stores the *IPv4* address of the node that has selected it as the *MPR.DestinaitonAddress* and *DestinationLocation* fields refer to the information of a destination node for which this node has been selected as *MPR*. *Cost* and *NodeCost* fields store the total cost (Eq. (1)) and link cost (Eq. (3)) between the selector node and the node itself. The cost calculation process is given in section 4.3.3. A node may update its *MPR_Table* once it receives a *TC* message from its neighbors. Since the source node cannot be selected as *MPR* node, its *MPRSelectorAddress* field always contains *NULL* value, or equivalently *"0.0.0.0"*. Initially, the source nodes create a separate tuple in their *MPR* table with *MPRSelectorAddress* = *"0.0.0.0"*. *MPR_Table* enables a node to know if it is a *MPR* node or not. This table, also, decides *TC* generation. A node runs Algorithm 4 in association with Eq. (1) to select the next *MPR* node using the table entries.

4.2.3 *Cost_Table* format

This table (Fig. 8) stores the next selected *MPR* information. For example, if a node *b* is selected as an *MPR* by node *a* for a particular destination node *c*, then *DestNodeAddress* and *NextNodeAddress* fields are populated by *c* and *b*, respectively. *Cost* field stores the cost-related information (Eq. (1)) for selecting the next *MPR* node. A node updates its *Cost_Table* utilizing the information stored in *Neighbor_Table* and *MPR_Table* using Algorithm 4.

4.3 Extended TC message format

This section introduces the modified *TC* message as given in Fig. 9. Only the *MPR* nodes generate *TC* messages containing the information stored in *Cost_Table*, *MPR_Table* and *Neighbor_Table*. A node shares its selected *MPR* set with its neighbor nodes through *TC*. A neighbor node receiving the *TC* message updates its *MPR_Table* if its *IPv4* address is piggybacked in this message. *TC* modification or extension increases its size; however, the demerit is counteracted by reducing the number of *MPR* nodes (and hence *TC* messages). The sender node shares its own *IPv4* address, and the *MPR* set through *MPRSelectorNodeAddress* and *MPRNodeAddress* fields, respectively. *DestinationNodeAddress* contains the address of the destination node for which *MPR* has been selected. *NodeCost* field contains the cost between the sender node and the selected *MPR* node, and *Cost* field contains the total cost to select an *MPR*.

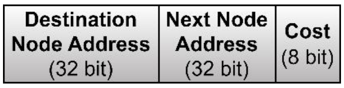

Figure. 8 *Cost_Table* format

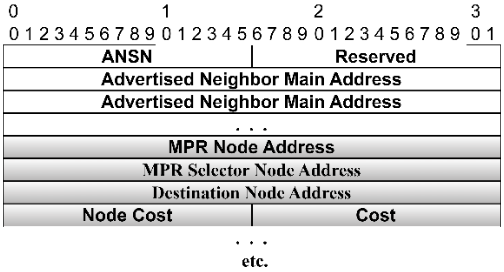

Figure. 9 Extended *TC* message format

4.3.1 Extended TC message format

TC message can only be generated and circulated by the selected *MPR* nodes and the source node. This can be implemented by checking the size of *MPR_Table* i.e. |*MPR_Table*| for each node. If |*MPR_Table*| ≠ ∅, only then it can send *TC* messages to its neighbors. A node can be identified as an *MPR* only if |*MPR_Table*| ≠ ∅ and *MPRSelectorAddress ≠ "0.0.0.0"*. *TC* messages are generated on basis of the information stored in *Neighbor_Table*, *MPR_Table*, and *Cost_Table*. The detailed *TC* message generation technique has been explained in Algorithm 2. This approach states that a node *y* checks its |*MPR_Table*| ≠ ∅ to generate the *TC* messages. For each tuple *i* of node *y*'s *Cost_Table*, the values of *NextNodeAddress*, *DestinationNodeAddress* and *Cost* fields are shared, respectively, through the *MPRNodeAddress*, *DestinationNodeAddress* and *Cost* fields of the generated *TC*. *MPRSelectorNodeAddress* field of *TC* contains the main address of node *y* and *NodeCost* represents the link cost. The remaining fields contain information following *RFC 3626* [11].

4.3.2 *TC* processing technique

Upon receipt of a *TC* message, a node *y* processes it only if its *IPv4* address is listed in the *MPRNodeAddress* field of the message. If the receiver node finds itself as listed, then it confirms itself to be an *MPR* node selected by the *TC* sending node and starts to process *TC* and updates its *MPR_Table*. Algorithm 3 shows the processing technique of the received *TC* message to update *MPR_Table*. For each row *i* of the received *TC*, a new tuple *j* is inserted into the node *y*'s *MPR_Table*. *MPRSelectorAddress*, *DestinationAddress*, *Cost*, *NodeCost* fields of each tuple *j* in *MPR_Table* of *y* stores the received information carried by *MPRSelectorNodeAddress*,*DestinationNodeAddress*, *Cost*, *NodeCost* fields, respectively, of each *i* of the received *TC*. *DestinationLocation* field of tuple *j* updates from node *y*'s *Dest_Table*. The remaining information is processed according to the basic *TC* message processing technique stated in [11].

4.3.3 Proposed cost function

The proposed *MPR* selection technique, illustrated in Algorithm 4, is based on the heuristic cost function presented in Eq. (1). For example, if *j* is selected as the next *MPR* of *i* for a particular destination *k*, then the cost for selecting *j* is the sum of the residual cost between *j* and *k* and node cost between *i* and *j*. It is assumed that the cost is directly proportional to Euclidean distance; the cost increases as the distance between two nodes increases. Euclidean distance between any two nodes is calculated as:

$$Cost^{\,j} = NodeCost^{\,i,j} + ResidualCost^{\,j,k} \tag{1}$$

$$D(p,q) = \sqrt{(q_x - p_x)^2 + (q_y - p_y)^2} \tag{2}$$

$$NodeCost^{i,j} = \frac{D(i,j)}{\alpha^j} \tag{3}$$

$$\alpha^j = 2 \times w^j + 1,\ w^j = willingness^j \tag{4}$$

$$ResidualCost^{j,k} = \frac{D(j,k)}{\beta} \tag{5}$$

$$Cost_{NextMPR^i} = \min_{\forall_{j\,\epsilon\,N(i)}} Cost^j \tag{6}$$

In Eq. (3), node cost represents the cost between any two 1-hop neighbor nodes. Node cost is directly proportional to the distance between these two nodes and inversely proportional to the willingness factor, *α*, of the reaching node. *α* is a function of willingness (Eq. (4)) of the neighbor node to forward a *TC* message. According to Eq. (3), if the willingness of neighbor node increases, node cost decreases, i.e., the possibility of being selected as *MPR* increases. On the other hand, node cost is high for a higher distance leading to a lesser possibility in *MPR* selection.

Residual cost (Eq. (5)) between the 1-hop neighbor *node(j)* and the destination *node(k)* is directly proportional to the Euclidean distance and inversely proportional to a normalization factor, *β*. If *D(j, k)* increases, it means that, node *j* is far away from destination *k*. This results in a lesser possibility

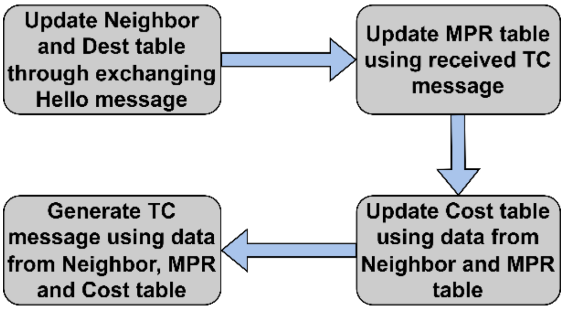

Figure. 10 The basic working process for calculating *MPR*

to select j as an *MPR* node for i. The normalization factor β depends on the nodes' transmission power and network area. In this study, β is determined heuristically.

Finally, the cost of the selected next *MPR* node of i is calculated using Eq. (6). Here, $N(i)$ represents all 1-hop neighbors of node i. From all the symmetric 1-hop neighbors of i, the selected next *MPR* is j, if the cost to reach j is lowest.

4.3.4 *MPR* calculation technique

A node calculates *MPR* periodically after each *TC_Interval* stated in the classical *OLSR*. Initially, the *MPR* is calculated according to the heuristic Algorithm 4, a node finds its next *MPR* set and updates *Cost_Table* to store *MPR* information as follows. A node finds its next *MPR* node based on a heuristic function stated in Eq. (1). The cost calculation process for selecting next *MPR* node follows Eq. (6).

If i and j represents each tuple of node y's *MPR_Table* and *Neighbor_Table* respectively, then it needs to find the lowest cost node j for each tuple i. Node y finds total cost for reaching each destination, stored in its *MPR_Table*, through each 1-hop neighbor j and finds the lowest cost neighbor j for each destination using Eq. (6). If y finds the lowest cost node p, from its all the 1-hop neighbors j, for a destination node q, then it considers node p as next *MPR* node for q. node y updates its *Cost_Table*'s *NextNodeAddress* and *DestinationNodeddress* fields with p and q respectively. *Cost* field contains the lowest cost for selecting p as next *MPR* node.

The basic working procedure is represented in Fig. 10. The neighbor table gets updated via continuous exchange of *Hello* messages. Each node can store neighbor information, including neighbor location and destination information, through exchanging *Hello* messages. Each node gets the information of the available destinations in the network as explained in section 4.2.1.

MPR_Table of a node is updated using the information piggybacked in the *TC* message. A node runs Algorithm 4 to find out the next *MPR* nodes based on the *Neighbor_Table* and *MPR_Table* tables. Each node with $|MPR\ Table| \neq \emptyset$ sends this *MPR*-related information to its neighbors using *TC* message. After receiving a *TC* message, a node can update its *MPR_Table* only if it is listed in this message.

Algorithm 2: *TC* message (*TC Msg*) generation

```
if $|MPR\_Table| \neq \emptyset$ then
  for i = 1, 2, .... do
    #i represents each tuple in Cost_Table.
    MPRNodeAddress (TC Msg) ← NextNodeAddress$^i$ (Cost_Table)
    MPRSelectorNodeAddress (TC Msg) ← SenderNodeAddress
    DestinationNodeAddress (TC Msg) ← DestinationNodeAddress$^i$ (Cost_Table)
    Cost (TC Msg) ← Cost$^i$ (Cost_Table)
    for j = 1, 2, .... do
      #j represents each tuple in MPR_Table.
      if DestinationNodeAddress$^i$ (Cost_Table) = DestinationNodeAddress$^j$ (MPR_Table) then
        for k = 1, 2, .... do
          #k represents each tuple in Neighbor_Table.
          if NextNodeAddress$^i$ (Cost_Table) = NeighborMainAddress$^k$ (Neighbor_Table) then
            NodeCost (TC Msg) ← NodeCost$^j$ (MPR_Table) + NodeCost$^k$ (Neighbor_Table)
            break
          end if
        end for
        break
      end if
    end for
  end for
end if
```

Algorithm 3: *TC* message (*TC Msg*) processing

```
for i = 1, 2, .... do
  #i represents each tuple in TC message.
  if ReceiverNodeAddress = MPRNodeAddress$^i$ (TC Msg) then
    MPRSelectorAddress (MPR_Table) ← SenderNodeAddress
    DestinationAddress (MPR_Table) ← DestinationNodeAddress$^i$ (TC Msg)
    Cost (MPR_Table) ← Cost$^i$ (TC Msg)
    NodeCost (MPR_Table) ← NodeCost$^i$ (TC Msg)
```

```
8:      for j = 1, 2, .... do
9:          #j represents each tuple in Dest_Table.
10:         if DestinationNodeAddress$^i$ (TC Msg) = DestinationNodeAddress$^j$ (Dest_Table) then
11:             DestinationLocation (MPR_Table) ← DestinationNodeLocation$^j$ (Dest_Table)
12:             break
13:         end if
14:     end for
15:     break
16:   end if
17: end for
```

Algorithm 4: Next *MPR* node Calculation

```
1: for i = 1, 2, .... do
2:   #i represents each tuple in MPR_Table.
3:   for j = 1, 2, .... do
4:       #j represents each tuple in Neighbor_Table.
5:       Calculate the Cost of each j node according to Eq.(1) and find out the minimum cost node k using Eq.(6)
6:   end for
7:   Insert the tuple of Cost_Table as below: Step 8-10
8:   DestinationNodeAddress (Cost_Table) = DestinationAddress$^i$ (MPR_Table)
9:   NextNodeAddress (Cost_Table) = NeighborMainAddress$^j$ (Neighbor_Table) which has been selected as k
10:  Cost (Cost_Table) =The Calculated Cost for reaching this node k
11: end for
```

Table 1. Simulation parameters

Platform used	Ubuntu-18.04
Type of network	*MANET*
Simulator used	*NS-3.30*
Simulation time	120 s
Total area	500*500 sq. m.
Number of nodes	50, 60, 70. 80, 90, 100
Transmit power	7.5 dBm
Mobility model	Random waypoint
Type of *MAC*	*IEEE 802.11b*
Transport layer	*UDP*
Total packet size	64 bytes
Pause Time	1, 5, 10 s
Stream index	0-9
Speed	4 m/s
Data rate	2048 bps
β	5

5. Simulation and results

5.1 Simulation parameters

Simulation experiments have been conducted using *NS3* (version 3.30) network simulator to validate our proposed *MPR* selection technique. Then, we compared the obtained results with standard *OLSR*. All simulation parameters have been summarized in Table 1.

5.2 Simulation results

Experiment results presented in this paper are taken as the average values after running the simulator 10 times for each scenario.

Fig. 11 demonstrates the comparison of the total number of selected *MPR* nodes between classical *OLSR* and proposed efficient *OLSR*. Experiment results show that, the selection of *MPR* nodes increases with increasing number of nodes, as, more nodes are needed to establish routes towards destinations. However, among all available nodes in the network, only a few nodes are selected as *MPR*s using our methodology. As, our proposed approach selects *MPR* from neighbor nodes using a heuristic cost function, only the nodes having less cost can be elected as *MPR*s for the particular destination nodes. Thus, all the optimal paths, established using the cost function stated in Eq. (1), towards each destination node, are composed of these selected *MPR* nodes. Consequently, all the necessary routes, needed for data forwarding, are being established with less number of selected *MPR* nodes. This scenario validates the thought that our proposed *MPR* selection technique outperforms the classical *OLSR*, *SSTB* and *M-OLSR* protocol in terms of 55% (on average), 28% (on average) and 49% (on average) less *MPR* selection respectively which causes less overhead or less propagation of *TC* messages.

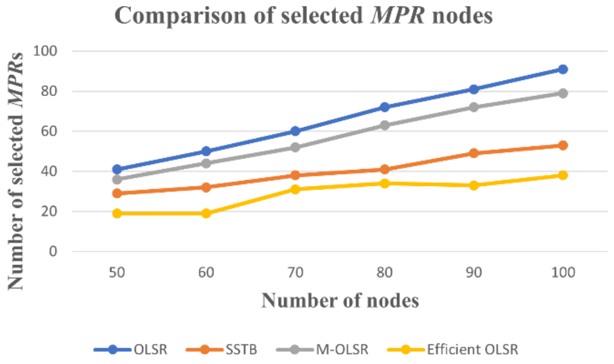

Figure. 11 Total selected *MPR* nodes

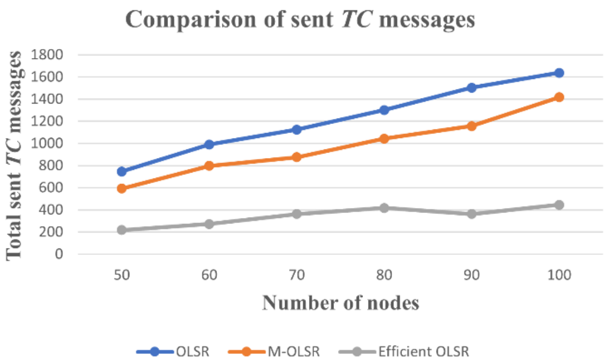

Figure. 12 Total sent *TC* messages

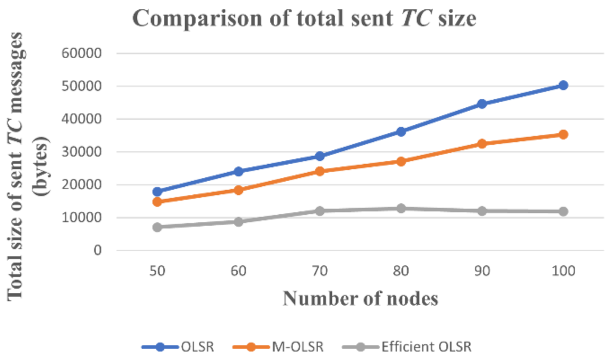

Figure. 13 Total size of sent *TC* messages

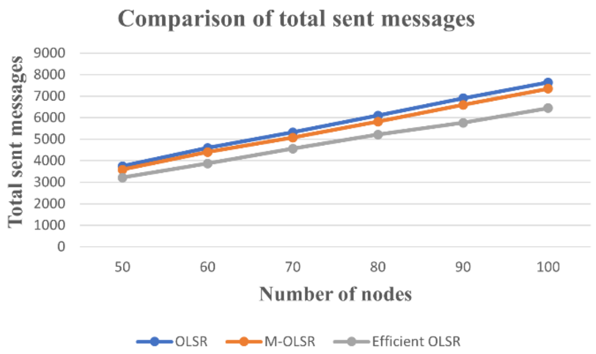

Figure. 14 Total sent messages

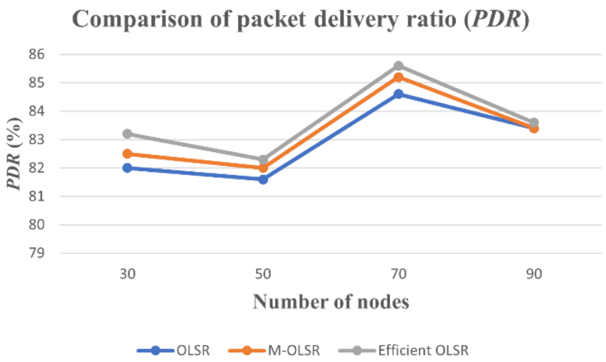

Figure. 15 Packet delivery ratio as a function of node number

Fig. 12 illustrates the total number of *TC* messages sent according to a different number of nodes both for standard *OLSR* and the proposed efficient *OLSR*. This result shows that *TC* dissemination increases according to the increasing node number for both protocols. Because, if number of node increases, it causes a rise in *MPR* selection. So, more *TC* messages are required to share network topology information. Moreover, the proposed method reduces the total *TC* dissemination for all cases. This is because, our proposed *OLSR* protocol selects less number of *MPR* nodes which absorb unnecessary *TC* flooding in the network. Consequently, our proposed protocol achieves up to 75% and 68% less *TC* propagation compared to the standard *OLSR* and *M-OLSR* protocol respectively.

Fewer *TC* dissemination also causes a reduction in the total size of the sent *TC* messages. This reduction in *TC* size is illustrated by Fig. 13. As the number of *MPR* nodes are reduced using the proposed protocol, it causes a reduction in the total number of flooded *TC* messages as well as *TC* size resulting less routing overhead.

Fig. 14 shows the comparison of total sent messages (*Hello* and *TC*) in the network. As, network density increases with higher number of nodes, number of sending messages also increases for establishing necessary routes. However, the experiment results show that our methodology produces up to 16% and 11% fewer messages than standard *OLSR* and *M-OLSR* respectively. Only *Hello* and *TC* messages are taken under consideration in calculating total messages for their significant impacts on routing overhead. This reduction in the total number of message dissemination causes less processing time as well as lower overhead.

On the other hand, packet delivery ratio increases with increasing pause time (Fig. 16). Because, if pause time increases, the possibility of link breaking

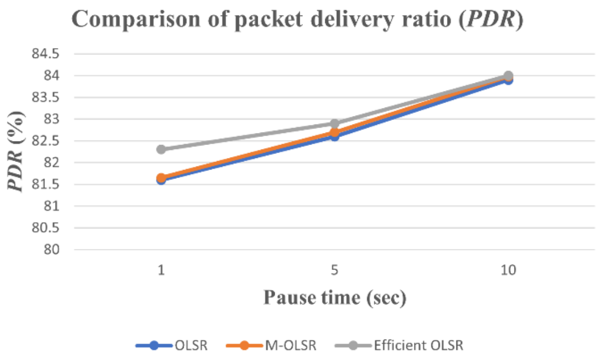

Figure. 16 Packet delivery ratio as a function of pause time

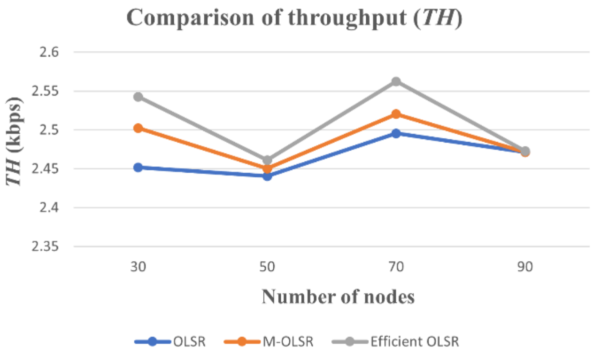

Figure. 17 Throughput as a function of node number

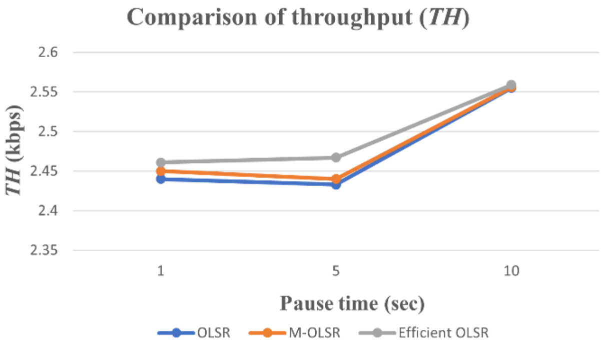

Figure. 18 Throughput as function of pause time

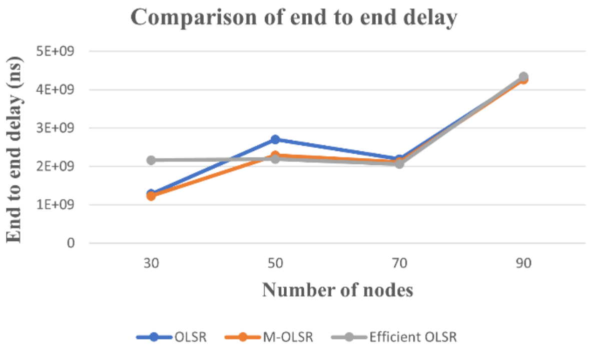

Figure. 19 Delay as function of node number

reduces, that, supports establishing optimal paths and increases packet delivery ratio.

Fig. 17 and Fig. 18 demonstrate the performance of the proposed *OLSR*, classical *OLSR* and *M-OLSR* in terms of throughput. These results depict that, throughput is being increased slightly in terms of both node number and pause time. From Fig. 18, it shows that, pause time creates more impacts on increasing throughput. Because, more stable links are established when pause time increases.

End-to-end delay is also compared in terms of node number in Fig. 19. Delay increases with increasing node number, as, the possibility of false *MPR* selection also increases. This causes establishing non-optimal routes which increases end-to-end delay for data transmission.

6. Conclusion

This paper proposes an improved *MPR* selection strategy for *OLSR* protocol to enhance its performance in terms of network overhead in *MANET*. The major contribution is to reduce the number of selected *MPR* nodes, which disseminates fewer *TC* messages without affecting the other performance matrices. The proposed *MPR* selection strategy requires additional repositories and header extensions of *Hello* and *TC* messages. The technique works according to a Euclidean distance-based heuristic function.

The experiment results show that routing overhead is reduced by 75% and 68% (as maximum) compared to the classical *OLSR* and *M-OLSR* protocols respectively. Our proposed *MPR* selection strategy also shows good performance compared to the standard *OLSR* and *M-OLSR* protocols in terms of packet delivery ratio, throughput and delay.

As the cost function is vital to the proposed *MPR* selection technique, in the future, the normalization and willingness factors and hence the cost function will be determined considering network area, node speed, and transmission power.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Md. Zahid Hassan, Shahid Md. Asif Iqbal and Asaduzzaman; methodology, Md. Zahid Hassan; software, Md. Zahid Hassan and Shahid Md. Asif Iqbal; validation, Md. Zahid Hassan, Shahid Md. Asif Iqbal, and Asaduzzaman; formal analysis, Md. Zahid Hassan and Asaduzzaman; investigation, Md. Zahid Hassan and Asaduzzaman; resources, Md. Zahid Hassan; data curation, Md. Zahid Hassan; writing—original draft preparation, Md. Zahid Hassan; writing—review and editing, Md. Zahid Hassan, Shahid Md. Asif Iqbal and Asaduzzaman; visualization, Md. Zahid Hassan and Asaduzzaman; supervision, Shahid Md. Asif Iqbal and Asaduzzaman; project administration, Md. Zahid Hassan, Shahid Md. Asif Iqbal and Asaduzzaman.

Acknowledgments

The authors received no external funding for this study.

References

[1] A. Agrawal, N. Faujdar, C. Romero, O. Sharma, G. Abdulsahib, O. Khalaf, R. Mansoor and O. Ghoneim, "Classification and comparison of ad hoc networks: A review", *Egyptian Informatics Journal*, Vol. 24, No. 1, pp. 1-25, 2023.

[2] R. Firmino, D. Mattos, D. Medeiros and P. Velloso, "MT-AODV:Resilient Communication in Military Mobile Ad-Hoc Networks through On-demand Multiple Routing Tables", In: *Proc. of 5th Conference on Cloud and Internet of Things (CIoT)*, Marrakech, Morocco, pp. 189-196, 2022.

[3] A. Arsalaan, H. Nguyen and M. Fida, "Impact of Bushfire Dynamics on the Performance of MANETs", In: *Proc. of 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS)*, Klosters, Switzerland, pp. 1-4, 2021.

[4] M. Hassan, S. Mostafa, H. Mahdin, A. Mustapha, A. Ramli and M. Jubair, "Mobile ad-hoc network routing protocols of time-critical events for search and rescue missions", *Bulletin of Electrical Engineering and Informatics*, Vol. 10, No. 1, pp. 192-199, 2021.

[5] A. Soomro, M. Fudzee, M. Hussain, H. Saim, G. Zaman, Atta-ur-Rahman, H. AlUbaidan and M. Nabil, "Comparative review of routing protocols in manet for future research in Disaster Management", *Journal of Communications*, Vol. 17, No. 9, pp. 734-744, 2022.

[6] V. Quy, V. Nam, D. Linh and L. Ngoc, "Routing algorithms for Manet-IOT Networks: A comprehensive survey", *Wireless Personal Communications*, Vol. 125, No. 4, pp. 3501-3525, 2022.

[7] J. Singh, G. Singh, D. Gupta, G. Muhammad and A. Nauman, "OCI-OLSR: An optimized control interval-optimized link state routing-based efficient routing mechanism for ad-hoc networks", *Processes*, Vol. 11, No. 5, pp. 1419, 2023.

[8] V. Quy, V. Nam, D. Linh, N. Ban and N. Han, "A survey of QoS-aware routing protocols for the Manet-WSN convergence scenarios in IOT Networks", *Wireless Personal Communications*, Vol. 120, No. 1, pp. 49-62, 2021.

[9] R. Sing and V. Narsimha, "Routing Protocols and Their Performance in Mobile Ad hoc Networks: A Quality of Service Optimization Perspective", In: *Proc. of OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON)*, Raigarh, Chhattisgarh, India, pp. 1-6, 2023.

[10] P. Shah and T. Kasbe, "A review on specification evaluation of broadcasting routing protocols in VANET", *Computer Science Review*, Vol. 41, No. C, pp. 100418, 2021.

[11] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", *RFC 3626*, https://www.rfc-editor.org/info/rfc3626, 2003.

[12] A. Eltahlawy, H. Aslan, E. Abdallah, M. Elsayed, A. Jurcut and M. Azer, "A survey on parameters affecting Manet Performance", *Electronics*, Vol. 12, No. 9, pp. 1956, 2023.

[13] R. Jisha, J. Joseph and M. Basheer, "Comprehensive Study on Mobile Ad-hoc Routing Protocols: OLSR, AODV and DSDV", In: *Proc. of IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, pp. 1-6, 2022.

[14] Q. Liu, X. Zhu, C. Zhou and C. Dong, "Advanced Fast Recovery OLSR Protocol for UAV Swarms in the Presence of Topological Change", In: *Proc. of 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Rio de Janeiro, Brazil, pp. 709-714, 2023.

[15] O. Sbayti, S. Ouhmi, B. Es-Sousy and K. Housni, "Enhancement in Quality of Routing Service in Vehicular ad-hoc Networks Based on the MPR Selection Algorithm of the OLSR Protocol", In: *Proc. of 6th International Conference on Advanced Communication Technologies and Networking (CommNet)*, Rabat, Morocco, pp. 1-6, 2023.

[16] Z. Wang, J. Zhang, Y. Zhang, H. Zhao and J. Wei, "Comparison of Mobile Ad Hoc Network Routing Protocols Based on NS3", In: *Proc. of IEEE 22nd International Conference on Communication Technology (ICCT)*, Nanjing, China, 2022, pp. 11-16, 2022.

[17] R. Ramamoorthy, R. Naidu, S. E and A. M, "Performance Investigation of Hybrid Bio-Inspired Zone Routing Protocol Over Proactive and Reactive Routing Protocols", In: *Proc. of International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Theni, India, pp. 1-6, 2023.

[18] Y. Hamzaoui, M. Chekour and M. Amnai, "Enhanced MPR Selection Algorithm Based on Metrics for the OLSR Routing Protocol", In: *Proc. of 6th International Conference on Advanced Communication Technologies and Networking (CommNet)*, Rabat, Morocco, pp. 1-6, 2023.

[19] L. Maccari, M. Maischberger and R. Cigno, "Where have all the MPRS gone? on the optimal

selection of Multi-Point Relays", *Ad Hoc Networks*, Vol. 77, pp. 69-83, 2018.

[20] L. Maccari and R. Cigno, "How to reduce and stabilize mpr sets in olsr networks", In: *Proc. of IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Barcelona, pp. 373-380, IEEE, 2012.

[21] D. Zhang, Y. Cui and T. Zhang, "New quantum-genetic based OLSR protocol (QG-OLSR) for mobile ad hoc network", *Applied Soft Computing*, Vol. 80, No. C, pp. 285-296, 2019.

[22] P. Kumar and S. Verma, "Implementation of modified OLSR protocol in AANETS for UDP and TCP environment", *Journal of King Saud University - Computer and Information Sciences*, Vol. 34, No. 4, pp. 1305-1311, 2022.

[23] P. Kumari and S. Sahana, "Swarm based hybrid ACO-PSO meta-heuristic (HAPM) for QoS multicast routing optimization in Manets", *Wireless Personal Communications*, Vol. 123, No. 2, pp. 1145-1167, 2021.

[24] S. Kaviani, B. Ryu, D. Kim, J. Kim, C. Spiker and B. Harnden, "DeepMPR: Enhancing Opportunistic Routing in Wireless Networks via Multi-Agent Deep Reinforcement Learning", *MILCOM IEEE Military Communications Conference (MILCOM)*, Boston, MA, USA, pp. 51-56, 2023.

[25] S. Gangopadhyay and V. Jain, "A Position-Based Modified OLSR Routing Protocol for Flying Ad Hoc Networks", *IEEE Transactions on Vehicular Technology*, Vol. 72, No. 9, pp. 12087-12098, 2023.

[26] W. Ahmed, M. Warip, W. Abduljabbar and M. Elshaikh, "WS-OLSR: Multipoint Relay Selection in VANET Networks using a Wingsuit Flying Search Algorithm", *International journal of Computer Networks & Communications*, Vol. 14, pp. 37-49, 2022.

[27] P. Rahul and B. Kaarthick, "Proficient link state routing in Mobile ad hoc network-based Deep Q-learning network optimized with chaotic bat swarm optimization algorithm", *International Journal of Communication Systems*, Vol. 36, No. 1, pp. e5324, 2023.

[28] F. Lakrami, M. Kamili, N. Elkamoun, H. Sounni and O. Labouidya, "A secure based trust model for Optimized Link State Routing protocol (OLSR)", In: *Proc. of 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Istanbul, Turkiye, pp. 1-5, 2023.