International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 – No. 6, August 2017 – www.ijais.org

Minimum Connected Dominating Set based Virtual Backbone Construction in OLSR Protocol

P. S. Vinayagam
Department of Computer Science,
Pondicherry University Community College,
Puducherry, India

ABSTRACT
In ad hoc wireless networks, nodes have limited energy and short transmission range. There are no pre-designated routers in these networks and hence routing cannot be done in the conventional way as done in wired networks. To facilitate routing in ad hoc wireless networks, some sort of backbone like structure needs to be built. One of the widely used routing protocols in ad hoc wireless networks is Optimized Link State Routing Protocol (OLSR), which makes use of Multipoint Relay sets (MPRs) to construct the virtual backbone. Various improvements have been proposed in the literature for the MPR selection scheme used in OLSR to improve its efficiency. In this paper, with the aim of enhancing the performance of OLSR, we propose an Improved OLSR (IOLSR) protocol, wherein Minimum Connected Dominating Set (MCDS) is used, instead of multipoint relay sets, to construct the virtual backbone. The performance of IOLSR is compared with that of OLSR in terms of the performance metrics – throughput, packet delivery ratio, end-to-end delay and size of the backbone. From the results, it is found that IOLSR protocol performs better than OLSR with respect to all these metrics.

General Terms
Routing Protocol

Keywords
OLSR, IOLSR, MCDS, MPR, ad hoc network, wireless network

1. INTRODUCTION
Wireless networks can be either infrastructure based or infrastructure-less. In infrastructure based wireless networks, Access Points act as routers for the nodes within their communication range. They also have the capability to connect to other wired networks. On the other hand, infrastructure-less wireless networks, also known as ad hoc networks, do not have pre-designated routers. In such networks, the nodes act as intermediaries (routers) to route packets among themselves. However, it is not possible for any single node to communicate with all the available nodes over a larger area, as the nodes often have limited energy and limited communication range. Hence, multi-hop routing is used in such networks for the packets to reach their destinations.

As there are no pre-designated routers, routing in ad hoc networks cannot be done in the conventional way as done in wired networks. Instead, some sort of backbone like structure needs to be built. As pointed out in Funke et al. [1], several routing schemes in ad hoc networks first establish a virtual backbone and then route messages via backbone nodes. Virtual backbone is a set of nodes that can help with routing [2]. The task of forwarding packets is restricted to a sub-set of nodes, which would form a routing backbone. Nodes that are not part of the backbone and wish to send a packet to another node in the network simply forward the packet to the nearest backbone node. It is the backbone nodes that are responsible for getting the packet to the destination [3].

Particularly, in the case of Optimized Link State Routing Protocol (OLSR), Multipoint Relay (MPR) sets serve as the virtual backbone. It is imperative that MPR be very efficient for OLSR to perform well. But, from reviewing the existing works on OLSR, it is identified that various improvements have been proposed for the MPR selection scheme used in OLSR to increase its efficiency.

In this paper, with the aim of enhancing the performance of OLSR, we propose a new Improved OLSR (IOLSR) protocol, wherein the MPR set is replaced with Minimum Connected Dominating Set (MCDS) to form the virtual backbone. As stated in Funke et al. [1], one common way of constructing a backbone is based on the construction of a minimum connected dominating set (MCDS). CDS has been extensively used for routing and broadcast in wireless ad hoc networks [4]. Simulations are carried out to compare the performance of the proposed IOLSR protocol with that of OLSR protocol with respect to the performance metrics, viz., throughput, packet delivery ratio, end-to-end delay and size of the backbone.

The rest of the paper is organized as follows. Section 2 consolidates existing work relating to modifying and improving the MPR selection method in OLSR, and work relating to construction of CDS. OLSR protocol and CDS are briefed in Section 3. The message formats and algorithms used for constructing virtual backbone in IOLSR using MCDS are discussed in Section 4. Section 5 presents the simulation results and section 6 concludes the paper.

2. RELATED WORK
The following sections consolidate the existing work relating to modifications done in the MPR selection scheme of OLSR in order to enhance its performance, and work relating to construction of CDS.

2.1 MPR Selection Scheme in OLSR
Prior research work has suggested various modifications in the MPR selection scheme of OLSR to improve its performance. Ge et al. [5] suggest modifications to OLSR with respect to MPR selection and routing table computation. They focus on supporting quality of service routing in OLSR and provide heuristics that allow OLSR to find the maximum bandwidth path.

In [6] Badis et al. use delay and bandwidth measurements to improve the quality requirements in MPRs selection. Li et al. [7] state that the greedy algorithm given in RFC 3626 has problems with MPR selection. To solve the problem, they propose an algorithm that decreases the number of MPRs and the number of TC packets. An energy aware MPR selection mechanism for OLSR routing protocol to improve its energy performance in mobile ad-hoc networks is provided in [8].

Koga et al. [9] state that OLSR selection process does not consider QoS requirements and provide schemes to select MPRs with high efficiency. A smaller set of MPRs that give better QoS paths between any two nodes is determined with the aim of maximizing the QoS effect for a given maintenance cost. Yamada et al. [10] show presence of redundant control messages in the MPR concept and present a cooperative MPR selection procedure that reduces the number of TC packets. An algorithm that eliminates the redundancy in terms of the total number of nodes selected as MPRs is presented in [11].

In [12] Boushaba et al. present an enhancement of the MPR selection algorithm in OLSR based on local databases of neighbor nodes extended to three hops. The enhancement is done to reduce the number of TC packets. Kitasuka and Tagashira [13] propose a method called shared MPR selection, for efficiently selecting MPR set in terms of reducing topology control traffic of OLSR. The proposed method shares MPRs between a node and its neighbor nodes. Kots and Kumar [14] state that the original MPR selection in OLSR does not consider quality metrics and thereby do not select quality nodes as MPR. They propose a fuzzy logic based routing metric that finds quality MPR. Other related works include [15-19].

2.2 Connected Dominating Set

Connected Dominating Set (CDS) has been used in numerous works for forming virtual backbone. The concept of virtual backbone was first proposed in Ephremides et al. [20]. Later, Guha and Khuller [21] proposed two approximation algorithms for the CDS problem. The first one is a greedy algorithm, for which efficient implementation is also provided. The second one is the improvement of the first algorithm. It involves finding a dominating set in the first phase and connecting the dominating set in the second phase.

Usage of MCDS as virtual backbone in ad-hoc networks has been studied in [22] and MCDS routing algorithm is proposed. Wu and Li [23] propose a distributed algorithm to construct CDS. They employ a marking process, where the marked vertices form a connected dominating set. Two distributed heuristics for constructing CDS are provided in [24], which needs only single-hop neighborhood information.

Wu et al. [25] provide a method of constructing power-aware CDS, based on a dynamic selection process, where a node with higher energy level is given preference. An algorithm for constructing MCDS is presented in [26]. It is a fully localized algorithm, wherein each node requires the knowledge of single hop neighbors and a constant number of two-hop and three-hop neighbors alone. A completely localized one-phase distributed algorithm for constructing CDS is proposed in [27].

Dai and Wu [28] have proposed a generalized dominant pruning rule to reduce the size of the dominating set. CDS construction algorithms that provide diameter reduced, risk reduced and interference aware dominating sets, without increasing CDS size have been proposed in [29]. Kamei and Kakugawa [30] present a self-stabilizing distributed approximation algorithm to construct minimum connected dominating set.

Kim et al. [2] provide algorithms to construct quality CDS in terms of size, diameter and Average Backbone Path Length (ABPL). Two centralized CDS construction algorithms and a distributed algorithm are proposed. The algorithms consider energy to extend network lifetime. Sheu et al. [31] propose a distributed algorithm to form a stable CDS, which forms a CDS by keeping a node with many weak links from being selected as a member of CDS.

A distributed local algorithm to compute CDS, where the nodes are assumed to have information about their locations, is proposed in Kassaei et al. [3]. Yin et al. [32] present a distributed single-phase CDS construction algorithm that constructs a CDS in a single phase using one-hop neighborhood information. Sakai et al. [4] propose timer-based CDS protocols, wherein a number of initiators are first elected and then, using timers, CDS is constructed from the initiators with the localized information.

An energy efficient CDS construction algorithm is presented in [33], where the node's mobility and residual energy are considered to create a stable MCDS. Ting-jun et al. [34] provide a distributed CDS construction algorithm, which gives priority to the nodes with more energy and closer nodes, while selecting the backbone nodes. Fu et al. [35] put forth a centralized algorithm for constructing CDS. In the first stage of the algorithm, MIS is constructed by using the local strategy and then in the second stage, the MIS nodes are connected by adding intermediate nodes to construct CDS. Other related works include [36-46].

From reviewing the existing literature it follows that prior research work has suggested various modifications in the MPR selection scheme of OLSR to enhance its performance. At the same time, CDS has also been largely used as the virtual backbone of wireless networks. This motivated us to construct the virtual backbone in OLSR using MCDS instead of MPR, with the aim of enhancing the performance of OLSR.

To this end, we propose a new improved OLSR protocol, wherein the MPR set is replaced with MCDS and its effect on network performance is examined. The protocol so proposed is termed as Improved OLSR (IOLSR) protocol, whose performance is compared with that of OLSR protocol in terms of the metrics: throughput, packet delivery ratio and end-to-end delay. Both the protocols are compared with respect to their backbone size also.

3. PRELIMINARIES

The working of the OLSR protocol and the concept of connected dominating set are briefly discussed in this section.

3.1 Optimized Link State Routing protocol

Optimized Link State Routing (OLSR) protocol, a proactive type of routing protocol, has been described in RFC 3626 [47]. This protocol uses Multipoint Relay sets to construct the virtual backbone. Nodes in the MPR set alone forward the control traffic. This way, the number of transmissions is reduced and flooding is controlled in the network.

OLSR protocol has many advantages over other protocols. It provides readily available routes to all the destinations in the network. The number of retransmissions needed to flood a message is reduced. Moreover, it requires only partial link state information to calculate the shortest path routes. In OLSR protocol, as the nodes periodically keep on sending the control messages, reliable transmission of control messages is ensured over time. Route calculation is done in each and every node in a distributed manner without depending on any central entity. Associated with each control message is a sequence number which frees OLSR from the burden of sequenced delivery of messages.

In OLSR, each node selects a set of nodes from among its 1-hop neighbor nodes as Multipoint Relay set. The nodes in the MPR set effectively cover all symmetric strict 2-hop neighbor nodes. The nodes that are not in the MPR set only receive and process broadcast messages. They do not retransmit those broadcast messages. This way flooding of messages is effectively minimized. Also, the MPR set must be smaller in size, to control flooding.

Apart from the MPR set, each node maintains a Multipoint Relay Selector Set. This set contains information about those nodes that have selected that particular node as their MPR. The MPR Selector Set is computed based on the information that is transmitted among nodes using periodic HELLO messages.

HELLO messages are periodically transmitted over all the available interfaces of a node. These messages are never forwarded. To accomplish this, Time To Live (TTL) of these messages is set to 1. HELLO messages serve three purposes: link sensing, neighbor detection and MPR selection signaling. Link sensing is used to generate the Link Set which describes all the links between the local interfaces and the neighbor interfaces. Neighbor detection is used to construct the 1-hop Neighbor Set. MPR selection signaling is used to populate the MPR selector set.

Topology Control (TC) messages are used to pass link-state information of each and every node to all the other nodes in the network. This enables the nodes to compute their routing table. The list of addresses given in the TC messages may be incomplete, but get updated over time. TC messages are sent using MPRs. They should be sent both in the case of new node insertions and link failures. Also, when the MPR selector set gets changed owing to link failure, TC message should be transmitted.

When a node is selected as a multipoint relay by any of its neighbors, it announces this information in the control messages at periodic intervals. As a result of this, nodes come to know about the reachability of other nodes in the network. Using this information, the routes are formed from a given node to various destinations.

3.2 Minimum Connected Dominating Set (MCDS)

In a given graph G = (V, E), a Dominating Set S of G is a subset of V: each node u in V is either in S or adjacent to some node v in S. In other words, a Dominating Set of a graph G = (V, E) is a set of nodes V′ such that $\forall(v, w) \in E$, $v \in V'$ or $w \in V'$. A Connected Dominating Set of G = (V, E) is a Dominating Set of G such that the subgraph of G induced by the nodes in this set is connected. The nodes in a CDS are called the dominators. The nodes other than the dominators are called the dominatees [2].

In a network, the hosts in CDS C can communicate with each other without using hosts in V – C [22]. The size of a CDS is equal to the number of dominators [2]. Among all CDSs of graph G, the one with minimum cardinality is called a minimum connected dominating set (MCDS) [46]. The problem of computing an MCDS is NP-hard [2].

Figure 1 shows a network with 8 vertices. Figure 2 shows the corresponding MCDS for the network. As shown in figure 2, only 3 nodes, viz., node 3, 4 and 6 are used to form the virtual backbone, which can be used to transmit messages to all the other nodes.

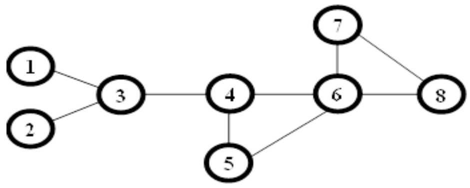

Fig 1: A sample network

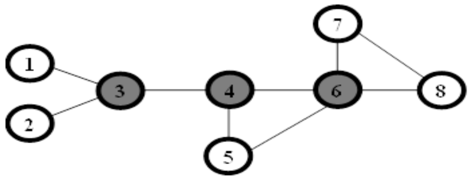

Fig 2: MCDS of the network shown in Fig 1

4. IMPROVED OPTIMIZED LINK STATE ROUTING (IOLSR) PROTOCOL

In the proposed IOLSR protocol, the virtual backbone is constructed using connected dominating set instead of multipoint relay set. The size of the virtual backbone obtained using CDS is smaller compared to the one obtained using MPR. As an example, consider the networks shown in figure 3 and figure 4.

The black coloured nodes in the network in figure 3 represent the virtual backbone constructed using MPR in OLSR. In the case of MPR as shown in figure 3, each node selects a few nodes as Multipoint Relay set nodes. These nodes are chosen from among its 1-hop neighbor nodes. The criterion used for selection of nodes as MPR nodes is effective coverage of all symmetric strict 2-hop neighbor nodes.

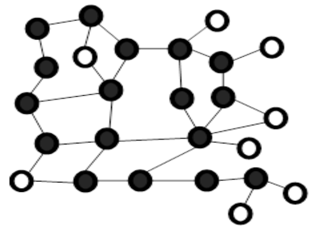

Fig 3: Virtual backbone using MPR

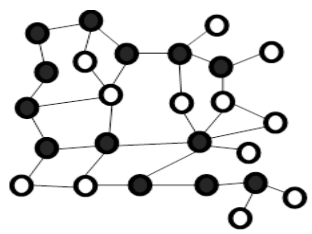

Fig 4: Virtual backbone using CDS

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 – No. 6, August 2017 – www.ijais.org

Figure 4 shows the virtual backbone constructed using CDS in IOLSR. The black coloured nodes are dominators and the white coloured nodes are dominatees. The black coloured nodes form the MCDS which acts as the virtual backbone of the network. All the dominatees are connected to at least one dominator in the network.

From figure 3 and figure 4, it follows that the virtual backbone obtained using CDS is smaller compared to the one constructed using MPR set. Hence, using CDS, there would be reduction in number of control messages that are being transmitted. This would in turn lower the number of collisions and provide high packet delivery rate.

In the following sections, we discuss the various algorithms and the message format that have been used in the IOLSR protocol.

4.1 Message Format

The format of the HELLO message in OLSR is as shown in figure 5. Changes have been made to the format of the HELLO message, as shown in figure 6, to suit the requirements of the proposed IOLSR protocol. The new fields that are added are Weight, Colour, HopToRoot, NodeID and DomNodeID. Weight of a node is a weighting function to denote the importance of the node based on its degree. Colour of a node could be white or black. HopToRoot gives the distance of the node to the root node in terms of number of hops. NodeID is a unique number used for identification of the node. DomNodeID contains the id of the node which needs to be upgraded as a dominator for a given node sending the HELLO request. As HELLO messages are never forwarded, their Time To Live (TTL) is set to 1.

HELLO messages are transmitted at regular interval among 1-hop neighbors. The information about neighbor nodes extracted from those messages is stored in neighbor tuple of a node. In OLSR, the neighbor tuple of a node contains the IP address of its neighbors, their status and willingness. The status indicates whether the link with the neighbor is symmetric or asymmetric. The willingness of a node describes the level of willingness of the node to forward traffic on behalf of other nodes. Willingness could be "Never", "Low", "Default", "High" or "Always". For the IOLSR protocol, the neighbor tuple of the node has been modified to additionally store the weight, colour, hop distance to root and node id of all the neighbor nodes.

Reserved		HTime	Willingness
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			
…			
…			
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			

Fig 5: OLSR HELLO message format [47]

Weight	Colour	HopToRoot	NodeID
Reserved	DomNodeID	HTime	Willingness
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			
…			
…			
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			

Fig 6: IOLSR HELLO message format

4.2 Algorithm

Upon reception of control messages like HELLO and TC, the IOLSR protocol uses the IOLSR_CDS algorithm to populate the routing table and form the virtual backbone. The neighbor node information received through HELLO messages and the link state information received through TC messages is used to populate the routing table using algorithm RT_Entries. The algorithm PCS_Computation then uses the routing table information to compute the parent, child and sibling relationship of a node with its neighbouring nodes. This information is in turn used by algorithm Dom_Computation to determine the colour of a node based on the colour information available about its neighbours in the form of parent set, child set and sibling set. The algorithms work in a distributed manner, using which each node determines its colour, either 'black' or 'white'. The black nodes are the members of the MCDS, and collectively they form the virtual backbone of the network and are used to transmit control messages.

Algorithm IOLSR_CDS()

1. Populate the routing table of the node using algorithm RT_Entries()
2. Compute the parent set, child set and sibling set of the node using algorithm PCS_Computation()
3. Determine the colour of the node using algorithm Dom_Computation()

End IOLSR_CDS

To start with, a node populates its routing table with route information to all destination nodes in the network. For this the node first stores the route information to all of its neighbors in the routing table. This is done via the neighbor set information that is received though the HELLO messages. Next, the route information to all other destination nodes (i.e. nodes other than the neighbor nodes) is stored. This reachability information to other destination nodes is determined using the link state information, which is disseminated among the nodes using Topology Control messages [47]. This complete process of populating the routing table in IOLSR is done using algorithm RT_Entries.

The routing table entries are used to determine the distance (i.e. number of hops) of a node to the root node. This hop to root information is then disseminated using HELLO messages. The PCS_Computation and Dom_Computation algorithms use this hop to root information.

Algorithm RT_Entries()

1. Populate Neighbor Set N(u) for a given node u, using HELLO messages received from 1-hop neighbor nodes
2. for each node v in N(u) do
3. Add reachability details of v to routing table

```
4.  end for
5.  Let h represent the hop count
6.  h = 1
7.  while (true) do
8.    for each entry x in the Topology Set T(u) do
9.      h = h + 1
10.     if the destination address in x is not available in the
        routing table then
11.       if there is a path to the next hop address for the
          given destination with hop distance h then
12.         add the route entry for that destination in
            the routing table with hop distance equal
            to h
13.       end if
14.     end if
15.   end for
16.   If no new entry has been added in the above steps
      (steps 8 through 15) then exit the loop
17. end while
```

End RT_Entries

After populating the routing table, the node computes the parent set, child set and sibling set using algorithm PCS_Computation. The parent, child or sibling relation of a node with other nodes is determined based on the distance of the nodes to the root. This distance information in terms of number of hops is disseminated among nodes via HELLO messages at periodic intervals. In the case of mobility of nodes, this information is subject to frequent change, based on the location of nodes.

The distRoot(x) refers to the distance of a given node ‘x’ to the root node measured in terms of number of hops. A node is chosen as the root node using a leader selection algorithm. The parent set of a node contains the list of those neighbor nodes which are nearer to the root as compared to the node under consideration. That is, their distance to root node in terms of number of hops is less than that of the node under consideration. Likewise, child set contains the list of those neighbor nodes which are farther away from the root as compared to the node under consideration. This is also based on the distance measured in terms of number of hops. The sibling set contains the list of those neighbor nodes that have the same distance to the root, in terms of number of hops, as that of the node under consideration.

Algorithm PCS_Computation()

1. Let u refer to the node under consideration
2. Let N(u) refer to the Neighbor Set of u
3. for each v in N(u) do
4.   if $dist_{Root}$ (v) < $dist_{Root}$ (u) then
5.   Add node v to P(u) where P(u) refers to the parent set of node u
6.   else if $dist_{Root}$ (v) > $dist_{Root}$ (u) then
7.   Add node v to C(u) where C(u) refers to the child set of node u
8.   else if $dist_{Root}$ (v) = $dist_{Root}$ (u) then
9.   Add node v to S(u) where S(u) refers to the sibling set of node u
10.   end if
11. end for

End PCS_Computation

Now the node proceeds to determine its colour based on the information obtained through algorithms RT_Entries and PCS_Computation. A node can either be ‘white’ or ‘black’ in colour. White nodes represent the dominatees and the nodes which are black represent the dominators. The collection of black nodes forms the MCDS.

To determine its colour, a node first checks for its adjacency to any dominator node. If it finds a dominator in its neighbor nodes, it becomes a dominatee. Otherwise, the node determines whether it needs to become a dominator to form a connected dominating set. If the node becomes a dominator, it looks for connectivity with the rest of the network via CDS and if required requests one of its parent node to become a dominator. The algorithm Dom_Computation carries out the above process of determining the colour of the node. It is based on the distributed CDS algorithm proposed in [2].

Algorithm Dom_Computation()

1. Let u be the current node under consideration
2. if node u is the root node then
3. u_{colour} = ‘black’
4.   return
5. end if
6. Let P(u) represent the parent set of node u
7. Let S(u) represent the sibling set of node u
8. Let $S_{undecided}(u) \subseteq S(u)$ represent the set containing the nodes who have not decided their colour
9. if (v_{colour} is decided $\forall$ v $\in$P(u)) and (u_{weight} is maximum in $\{S_{undecided}(u)\} \cup \{u\}$) then
10.   if colour of any one of the nodes in P(u) is ‘black’ then
11. u_{colour} = ‘white’
12.   end if
13. end if
14. Let $S_{mweight}(u) \subseteq S(u)$ represent the set containing the nodes whose weight is greater than u_{weight}
15. if (v_{colour} is ‘white’ $\forall$ v $\in$P(u)) and (x_{colour} is ‘white’ $\forall$ x $\in S_{mweight}(u)$) then
16. u_{colour} = ‘black’ provided it is not a leaf node which is determined using child set
17.   Find node y in P(u) where y_{weight} is maximum in P(u)
18.   Node y is requested to become a dominator through the HELLO message by setting the node id of y in DomNodeId field of the HELLO message
19. end if
20. if a node finds its id in DomNodeID of HELLO message, it colours itself ‘black’

End Dom_Computation

The above algorithms work in a distributed manner and each node determines its colour using only neighbour node information. The resulting set of black nodes forms the MCDS that acts as the virtual backbone of any given network.

5. SIMULATION AND RESULT ANALYSIS

NS3 [48], a discrete event network simulator, has been used to evaluate the performance of IOLSR. Simulations have been carried out for a wireless network of 50 nodes moving in a 1000m x 1000m area. The random waypoint mobility model is used and nodes move at a speed of 5 metres per second. Simulation is run for a total of 1000 seconds. To extract average values, each scenario was randomly simulated 10 times. In the following sections, performance metrics and the simulation results are discussed.

5.1 Performance Metrics

The IOLSR and OLSR protocols are compared with respect to throughput, packet delivery ratio and end-to-end delay. Also, the size of the backbone in both the protocols is compared. A brief discussion of these metrics follows.

5.1.1 Throughput
Throughput refers to the average rate at which data packet is delivered successfully from one node to another. A higher value of throughput implies better performance. It is usually measured in bits per second (bps).

Throughput = (P * S * 8) / T

where P is the number of packets successfully delivered, S is the size of the packet and T is the total duration of the simulation. A higher value throughput is always preferred of a routing protocol.

5.1.2 Packet Delivery Ratio
Packet Delivery Ratio (PDR) is the ratio of data packets received by the destination to those generated by the source.

Packet Delivery Ratio = R/S

where, R is the total number of packets received by the destination and S is the total number of packets generated by the source. A greater value of PDR implies better performance of the routing protocol.

5.1.3 End-To-End Delay
End-to-end delay measures the time taken by a data packet to reach its destination. This includes the delay that is incurred during route discovery, transmission delays and queuing in transmission.

End-to-end Delay = R – S

where R represents the time at which first data packet arrives at the destination and S denotes the time at which first data packet was sent by the source. The average end-to-end delay is calculated by dividing the sum of the time spent to deliver packets for each destination by the number of packets received by all the destination nodes. For end-to-end delay, the data packets that successfully reached the destinations alone are considered. A low end-to-end delay value implies better performance.

5.2 Simulation Results
The findings from the simulation results of IOLSR and OLSR protocols with respect to the metrics: throughput, packet delivery ratio and end-to-end delay, and the number of backbone nodes are discussed below.

The results of throughput pertaining to IOLSR and OLSR protocols are presented in figure 7. The x-axis represents the simulation time in seconds. The throughput measured in kilobits per second (kbps) is taken along the y-axis. It is evident from the figure that the throughput of IOLSR is considerably better than that of OLSR. From the start of the simulation, IOLSR exhibits good throughput which gradually increases and stabilizes over a period of time. The throughput of OLSR is almost stable throughout the entire simulation, but it is lower than that of IOLSR. A higher throughput implies better performance of the routing protocol. Hence, IOLSR is able to provide better network performance in terms of throughput as compared to OLSR protocol.

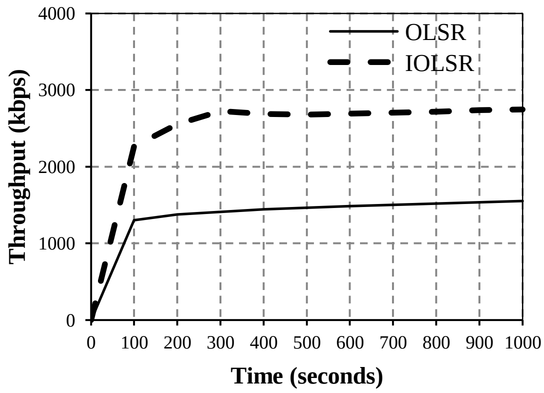

Fig 7: Throughput of IOLSR and OLSR

Figure 8 depicts the simulation results of packet delivery ratio of both IOLSR and OLSR protocols. From the figure, it is evident that the packet delivery ratio achieved by IOLSR is higher than that of OLSR. At the beginning of the simulation, the packet delivery ratio of both IOLSR and OLSR is higher as compared to the rest of the simulation. This is because, at the start of simulation, the number of control packets is minimum. As the simulation proceeds and the number of control packets increases, the packet delivery ratio gradually decreases and stabilizes over the run of the simulation. However, during the entire simulation, IOLSR is able to deliver more number of packets compared to OLSR. The better PDR of IOLSR as compared to OLSR could be attributed to the reduction in the number of dominators, as shown in figure 11, and hence reduction in the collision. In the case of OLSR, due to the effect of collisions, routing table may not be updated correctly always. This in turn would result in the decrease of packet delivery ratio. Thus, in terms of network performance with respect to packet delivery ratio, IOLSR is better than OLSR protocol.

The end-to-end delay of IOLSR and OLSR protocols is compared in figure 9. Initially, at the start of simulation, the delay in case of OLSR drops and stabilizes for some time of the simulation. Later, it gradually keeps on increasing over the run of simulation. In case of IOLSR, delay shows gradual increase for the entire duration of the simulation. On comparing the delay of IOLSR and OLSR during the entire simulation, it is observed that the end-to-end delay experienced in case of IOLSR is lower than that in OLSR. A low end-to-end delay value is always preferred of a routing protocol. IOLSR, therefore, has better network performance than OLSR, as it delivers packets with lower end-to-end delay.

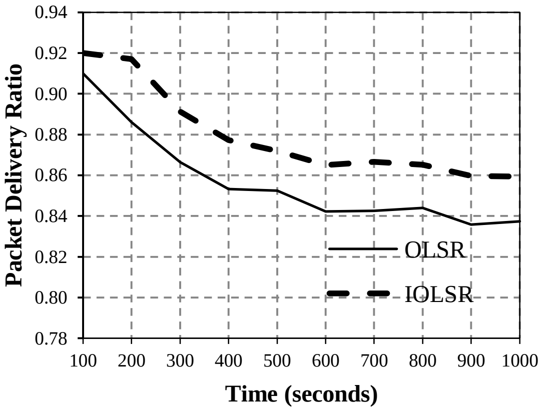

Fig 8: Packet Delivery Ratio of IOLSR and OLSR

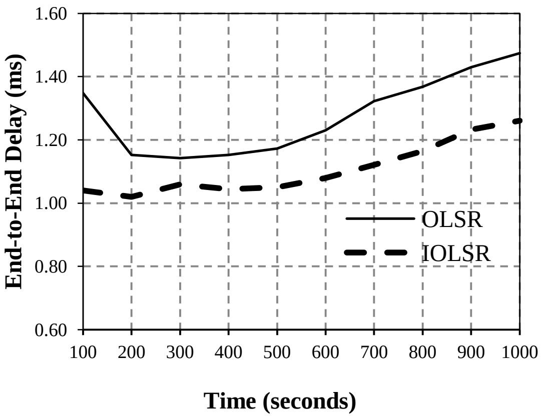

Fig 9: Delay in IOLSR and OLSR

A comparative chart showing the packet delivery ratio and delay of both IOLSR and OLSR protocols is depicted in figure 10. It is observed that in both the protocols the packet delivery shows gradual decrease with gradual increase in delay.

The size of the backbone in IOLSR protocol is compared with that of OLSR in figure 11. It shows the comparison between the number of dominator nodes in IOLSR and the number of MPR nodes in OLSR. From the figure, it is evident that the number of backbone nodes in IOLSR is less as compared to that of OLSR. The size of the virtual backbone should be kept to a minimum, as the number of control messages that are generated are comparatively lesser when the virtual backbone size is small. Hence, in terms of backbone size, the performance of IOLSR is better than OLSR protocol.

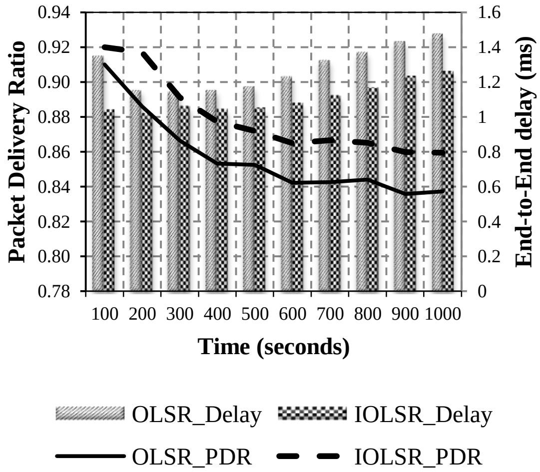

Fig 10: Comparative chart of PDR and Delay

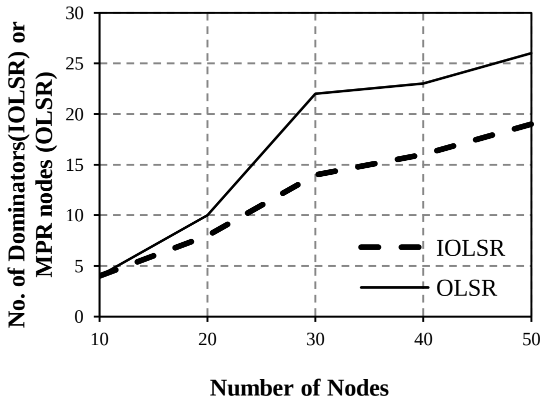

Fig 11: Comparison of backbone size in IOLSR and OLSR

6. CONCLUSION

In this paper, an Improved OLSR (IOLSR) protocol has been proposed, wherein MCDS is used instead of MPRs to construct the virtual backbone. IOLSR effectively replaces the MPRs of OLSR by MCDS. The message formats of OLSR were modified as per the requirements of the IOLSR protocol. The performance of IOLSR was evaluated against OLSR in terms of performance metrics: throughput, packet delivery ratio and end-to-end delay. Also, both the protocols were compared in terms of their backbone size. From the simulation results, it is found that IOLSR protocol performs better than OLSR protocol with respect to all these metrics. IOLSR protocol provides better throughput, higher packet delivery ratio and lower delay than OLSR. The size of the backbone in IOLSR protocol is smaller compared to that of OLSR.

7. REFERENCES

[1] Funke, S., Kesselman, A., Meyer, U., Segal, M. (2005). A simple improved distributed algorithm for minimum CDS in unit disk graphs. *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, (WiMob'2005),* vol. 2, pp. 220 – 223.

[2] Kim, D., Wu, Y., Li, Y., Zou, F., and Du, D. (2009). Constructing minimum connected dominating sets with bounded diameters in wireless networks. *IEEE Transactions on Parallel and Distributed Systems,* vol.

20, no. 2, pp. 147-157.

[3] Kassaei, H., Mehrandish, M., Narayanan, L., Opatrny, J. (2009). A new local algorithm for backbone formation in ad hoc networks, Proceeding. PE-WASUN'09 Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, Tenerife, Canary Islands, Spain, pp. 49-57.

[4] Sakai, K., Huang, S. C. H., Ku, W., Sun, M., Cheng, X. (2011). Timer-based CDS construction in wireless ad hoc networks. *IEEE Transactions on Mobile Computing,* vol. 10, no. 10, pp. 1388-1402.

[5] Ge, Y., Kunz, T., Lamont, L. (2003). Quality of service routing in ad-hoc networks using OLSR. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03).*

[6] Badis, H., Munaretto, A., Agha, K. A., Pujolle, G. (2004). Optimal path selection in a link state QoS routing protocol. *IEEE 59th Vehicular Technology Conference, VTC,* vol. 5, pp. 2570 – 2574.

[7] Li, Z., Yu, N., Deng, Z. (2008). NFA: A new algorithm to select MPRs in OLSR. *4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM '08*, Dalian, pp. 1 – 6.

[8] Rango, F. D., Fotino, M., Marano, S. (2008). EE-OLSR: energy efficient OLSR routing protocol for mobile ad-hoc networks. *IEEE Military Communications Conference, MILCOM 2008*, San Diego, CA, pp. 1 – 7.

[9] Koga, T., Tagashira, S., Kitasuka, T., Nakanishi, T., Fukuda, A. (2009). Highly efficient multipoint relay selections in link state QoS routing protocol for multi-hop wireless networks. *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops, WoWMoM 2009*, Kos, pp. 1-9.

[10] Yamada, K., Itokawa, T., Kitasuka, T., Aritsugi, M. (2010). Cooperative MPR selection to reduce topology control packets in OLSR. *TENCON 2010 - 2010 IEEE Region 10 Conference,* Fukuoka, pp. 293 – 298.

[11] Kitasuka, T., Tagashira, S. (2011). Density of multipoint relays in dense wireless multi-hop networks. *Second International Conference on Networking and Computing, (ICNC),* Osaka, pp. 134 – 140.

[12] Boushaba, A., Benabbou, A., Benabbou, R., Zahi, A., Oumsis, M. (2012). Optimization on OLSR protocol for reducing topology control packets. *2012 International Conference on Multimedia Computing and Systems (ICMCS),* Tangier, pp. 539 – 544.

[13] Kitasuka, T., Tagashira, S. (2013). Finding more efficient multipoint relay set to reduce topology control traffic of OLSR. *2013 IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM),* Madrid, pp. 1-9.

[14] Kots, A., Kumar, M. (2014). The fuzzy based QMPR selection for OLSR routing protocol. *Wireless Networks,* vol. 20, no. 1, pp. 1-10.

[15] Mans, B., Shrestha, N. (2004). Performance evaluation of approximation algorithms for multipoint relay selection. *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop*, pp. 480-491.

[16] Belkheir, M., Qacem, Z., Bouziani, M., Ghelamellah, A. (2012). An energy optimization algorithm for mobile ad hoc network. *International Journal of Soft Computing and Software Engineering (JSCSE),* vol.2, no.10, pp. 20-26.

[17] Malik, R. F., Rahman, T. A., Ngah, R., Hashim, S. Z. M. (2012). The new multipoint relays selection in OLSR using particle swarm optimization. *TELKOMNIKA,* vol.10, no.2, pp. 343-352.

[18] Maccari, L., Cigno, R. L. (2012). How to reduce and stabilize MPR sets in OLSR networks. *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),* Barcelona, pp. 373 – 380.

[19] Dashbyamba, N., Wu, C., Ohzahata, S., Kato, T. (2013). An improvement of OLSR using fuzzy logic based MPR selection. *15th Asia-Pacific Network Operations and Management Symposium (APNOMS),* Hiroshima, Japan, pp. 1 – 6.

[20] Ephremides, A., Wieselthier, J. E., Baker, D. J. (1987). A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE,* vol. 75, no. 1, pp. 56-73.

[21] Guha, S., Khuller, S. (1998). Approximation algorithms for connected dominating sets. *Algorithmica,* vol. 20, no. 4, pp. 374–387.

[22] Das, B., Bharghavan, V. (1997). Routing in ad-hoc networks using minimum connected dominating sets. *IEEE International Conference on Communications,* ICC '97 Montreal, Towards the Knowledge Millennium, vol. 1, Montreal, Que, pp. 376 – 380.

[23] Wu, J., Li, H. (2001). A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems,* vol. 18, no. 1–3, pp. 13–36.

[24] Alzoubi, K. M., Wan, P., Frieder, O. (2002). Distributed heuristics for connected dominating sets in wireless ad hoc networks. *Journal of Communications and Networks,* vol.4, no.1, pp. 22-29.

[25] Wu, J., Dai, F., Gao, M., Stojmenovic, I. (2002). On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *Journal of Communications and Networks,* vol.4, no.1, pp. 59-70.

[26] Alzoubi, K. M., Wan, P., Frieder, O. (2002). Message-optimal connected dominating sets in mobile ad hoc networks. *Proceeding MobiHoc'02 Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing,* EPFL Lausanne, Switzerland, pp. 157 – 164.

[27] Li, Y., Zhu, S., Thai, M., Du, D. (2004). Localized construction of connected dominating set in wireless networks. *In Proceedings of US National Science Foundation Proc. NSF International workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor and Peer-to-Peer Networks (TAWN'04),* Chicago, USA.

[28] Dai, F., Wu, J. (2004). An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems,* vol. 15, no. 10, pp. 908-920.

[29] Mohammed, K., Gewali, L., Muthukumar, V. (2005). Generating quality dominating sets for sensor network. *Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05),* pp. 204 – 211.

[30] Kamei, S., Kakugawa, H. (2007). A self-stabilizing distributed approximation algorithm for the minimum connected dominating set. *IEEE International Parallel and Distributed Processing Symposium,* IPDPS 2007, Rome, pp. 1 – 8.

[31] Sheu, P., Tsai, H., Lee, Y., Cheng, J. (2009). On calculating stable connected dominating sets based on link stability for mobile ad hoc networks. *Tamkang Journal of Science and Engineering,* vol. 12, no. 4, pp. 417-428.

[32] Yin, B., Shi, H., Shang, Y. (2011). An efficient algorithm for constructing a connected dominating set in mobile ad hoc networks. *Journal of Parallel and Distributed Computing,* vol. 71, no. 1, pp. 27–39.

[33] Chakradhar, P., Yogesh, P. (2013). Energy efficient minimum connected dominating set algorithm for MANETs. *IEEE International Conference on Recent Trends in Information Technology (ICRTIT),* Chennai, pp. 270 – 276.

[34] Ting-jun, S., Xu, S., Xu-ming, F. (2014). A Virtual Backbone Construction Algorithm Based on Connected Dominating Set in Wireless Sensor Networks. *International Conference on Computer, Communications and Information Technology (CCIT 2014),* Beijing, pp. 156-159.

[35] Fu, D., Han, L., Liu, L., Gao, Q., Feng, Z. (2015). An Efficient Centralized Algorithm for Connected Dominating Set on Wireless Networks. *Procedia Computer Science,* vol. 56, pp. 162 – 167.

[36] Das, B., Sivakumar, R., Bharghavan, V. (1997). Routing in ad hoc networks using a spine. *Proceedings. Sixth International Conference on Computer Communications and Networks,* Las Vegas, pp. 34 – 39.

[37] Sivakumar, R., Das, B., Bharghavan, V. (1998). The clade vertebrata: spines and routing in ad hoc networks. *Proceedings. Third IEEE Symposium on Computers and Communications, ISCC '98,* Athens, pp. 599 – 605.

[38] Stojmenovic, I., Seddigh, M., Zunic, J. (2002). Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems,* vol. 13, no. 1, pp. 14-25.

[39] Wan, P., Alzoubi, K. M., Frieder, O. (2004). Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications,* vol. 9, no. 2, pp. 141–149.

[40] Ruan, L., Du, H., Jia, X., Wu, W., Li, Y., Ko, K. (2004). A greedy approximation for minimum connected dominating sets. *Theoretical Computer Science,* vol. 329, no. 1-3, pp. 325 – 330.

[41] Kim, B., Yang, J., Zhou, D., Sun, M. (2005). Energy-aware connected dominating set construction in mobile ad hoc networks. *Proceedings of 14th International Conference on Computer Communications and Networks, ICCCN 2005,* pp. 229 – 234.

[42] Zongkai, Y., Dasheng, Z., Yuming, W., Jianhua, H. (2005). An efficient distributed algorithm for connected dominating set construction in wireless sensor networks. *Journal of Electronics (China),* vol. 22, no. 6, pp. 671-675.

[43] Cheng, X., Ding, M., Du, D. H., Jia, X. (2006). Virtual backbone construction in multihop ad hoc wireless networks. *Wireless Communications and Mobile Computing,* vol. 6, no. 2, pp. 183–190.

[44] Han, B., Zhang, L., Jia, W. (2010). Connected dominating set for topology control in ad hoc networks, in *Mobile Intelligence*. New Jersey: Wiley, ch. 2, pp. 26-42.

[45] Gao, B., Ma, H., Yang, Y. (2005). A new distributed approximation algorithm for constructing minimum connected dominating set in wireless ad hoc networks, in *Parallel and Distributed Processing and Applications*. Berlin Heidelberg: Springer, pp. 352–356.

[46] Cardei, M., Cheng, X., Cheng, X., Du, D. (2002). Connected domination in multihop ad hoc wireless networks. *Proceedings of the 6th Joint Conference on Information Science (JCIS),* North Carolina, pp. 251-255.

[47] Clausen, T., Jacquet, P. (2003). Optimized Link State Routing Protocol (OLSR), Request for Comments: 3626. Available: https://www.ietf.org/rfc/rfc3626.txt.

[48] NS3, http://www.nsnam.org.