A Fault-tolerant Improved OLSR Protocol using k-Connected m-Dominating Set

Vinayagam P. S.
Department of Computer Science, Pondicherry University Community College, Puducherry, India
E-mail: psvinayagam@gmail.com

Abstract—The inherent properties of ad hoc networks such as limited energy, short transmission range and absence of routers along with node mobility, node failures and link failures make routing a challenging task. In order to facilitate routing, virtual backbone has been proposed as a viable solution in the literature. Optimize Link State Routing (OLSR) protocol, a proactive routing protocol, uses Multipoint Relay (MPR) set to construct virtual backbone. Prior research has, however, identified various issues with the MPR selection scheme that needs improvement. One of the alternatives that could be used to construct virtual backbone is Connected Dominating Set (CDS). Although CDS generates a smaller virtual backbone, its 1-connected 1-domination nature may render a virtual backbone obsolete in case of networks which witness frequent node mobility, node failures and link failures. To overcome this, k-Connected m-Dominating CDS (kmCDS) could be used to construct fault- tolerant virtual backbone structure. In this direction, the present paper proposes a Fault-Tolerant Improved Optimized Link State Routing (FT-IOLSR) protocol that uses kmCDS to form fault-tolerant virtual backbone, effectively replacing the MPR set of OLSR protocol. Simulations are carried out to assess the performance of the FT-IOLSR protocol in relation to the OLSR protocol, with respect to various node speed and pause time combinations, and varying network size. The results show that the FT-IOLSR protocol is better in terms of packet delivery ratio under varying mobility and varying network size. Also it has been observed that, with increase in k-connectivity and m-domination factor, there is improvement in the performance of the protocol.

Index Terms—kmCDS, MPR, OLSR, Virtual backbone, Fault-tolerant, FT-IOLSR.

I. INTRODUCTION

Ad hoc networks are characterized by limited energy, short transmission range and absence of routers. Lack of access points in these networks puts the onus of routing on the participating nodes. Moreover, due to limited transmission range, a node cannot communicate with far-off nodes directly. Hence, multihop routing is used by a node to communicate with other nodes that are not within its transmission range. Also, lack of central administration makes self-organization of nodes essential for routing. Further, the mobility of nodes, node failures and link failures which are a common phenomenon in mobile ad hoc networks affect routing. These features which are inherent in ad hoc networks make routing a challenging task.

Flooding of messages can be used as a solution for routing. But this results in the broadcast storm problem, which occurs due to increased contentions and collisions in networks. To solve this, some sort of backbone like structure needs to be built, similar to the infrastructure in wired networks. To this end, literature suggests the construction of virtual backbone as the routing infrastructure of ad hoc networks.

The concept of virtual backbone was first proposed in [1]. In a network employing virtual backbone, a node in the network can either be a backbone node or not. Any non-backbone node has to be adjacent to at least one backbone node. Moreover, the set of backbone nodes has to be connected. Then, the routing path search space for each message is reduced from the whole network to the set of backbone nodes [2]. Routing messages are not broadcast to all the nodes. Instead backbone nodes alone exchange the routing messages. Whenever a non-backbone node wants to communicate to a destination node, it forwards the message to a neighbour backbone node. The backbone nodes in turn send the message to the destination node.

Optimized Link State Routing (OLSR) protocol is one such protocol that relies on virtual backbone structure for routing. It is a proactive routing protocol that uses Multipoint Relay (MPR) set to construct the virtual backbone. The performance of the protocol is largely dependent upon the MPR set. Existing research has, however, identified various issues in the MPR selection scheme of OLSR protocol that needs improvement. An attractive alternative to MPR is Connected Dominating Set (CDS) which is a well-known solution for constructing virtual backbone [3].

CDS can be used to replace MPR set to construct virtual backbone in OLSR protocol. However, in networks that witness frequent node mobility, node failures and link failures, even the virtual backbone constructed using CDS can fast become obsolete, as it offers only 1-connected 1-domination. In the event of

failure of even a single backbone node, the virtual backbone is prone to break which ultimately breaks the network. Hence, it is essential that a fault-tolerant virtual backbone is constructed which can withstand node and link failures. For this, k-Connected m-Dominating Set (kmCDS) could be employed.

In this direction, the present paper proposes Fault-Tolerant Optimized Link State Routing (FT-IOLSR) protocol that uses kmCDS to construct fault-tolerant virtual backbone, effectively replacing MPR set in the OLSR protocol. The performance of the FT-IOLSR protocol is examined in relation to that of the OLSR protocol through simulations.

The rest of the paper is organized as follows. Section 2 presents an overview of the OLSR protocol followed by the concept of CDS. Section 3 summarizes the prior research work related to the construction of kmCDS. The formation of fault-tolerant virtual backbone in FT-IOLSR protocol is elaborated in section 4. The simulation results comparing the performance of FT-IOLSR protocol in relation to that of the OLSR protocol are presented in section 5. Section 6 concludes the paper.

II. Overview

In this section, OLSR protocol is discussed briefly. The definitions and the basic concepts of CDS are also presented.

A. Optimized Link State Routing (OLSR) Protocol

OLSR is a proactive routing protocol for mobile ad hoc networks described in RFC 3626 [4]. OLSR protocol uses MPR set as the virtual backbone. Each node selects a set of nodes in the symmetric 1-hop neighbourhood to retransmit its messages. These set of selected neighbour nodes are called the MPR set of the node. The MPR nodes alone forward the control traffic. The nodes which are not part of the MPR set only receive and process broadcast messages. They do not retransmit broadcast messages. The MPR set is chosen in such a way that it covers all the symmetric strict 2-hop nodes.

Multipoint Relay Selector set of a node contains information about the set of neighbours that have selected the node as MPR. This information is gathered via periodic HELLO messages received from the neighbours. HELLO messages take care of three tasks: link sensing, neighbour detection and MPR selection signaling. Nodes broadcast HELLO messages to their neighbours and the HELLO messages are never forwarded [4].

Each and every MPR node broadcasts topology control (TC) messages to create the topology information base. The TC messages are flooded to all the network nodes and the information transmitted via the TC messages enables the nodes to populate their routing table [4]. The routing table enables the nodes to route data to other nodes in the network.

B. Connected Dominating Set (CDS)

The history of dominating set can be traced back to 1850's [5]. Given a graph G = (V,E), a Dominating Set (DS) of G is a subset C ⊂ V such that each node either belongs to C or is adjacent to at least one node in C [6]. In other words, a Dominating Set of a graph G = (V,E) is a set of nodes V′ such that ∀(v,w) ∈ E, v ∈ V′ or w ∈ V′ [7].

A Dominating Set is connected if there exist a path between any two nodes in the set and the path only consists of the nodes in the set [8]. 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. Each dominatee is dominated by a dominator [9]. The size of a CDS is equal to the number of dominators [7]. Among all CDSs of graph G, the one with minimum cardinality is called a Minimum Connected Dominating Set (MCDS) [10].

In CDS based routing, the dominator nodes alone maintain the routing information. A dominatee node in order to send a message to another dominatee, will send it to its dominator. Then the search space for the route is reduced only within the CDS. That is, in a CDS C, the nodes in C can communicate with any other node in the same set without using nodes in V – C [11]. When the message reaches the destination's dominator, the message is delivered to the destination via the said dominator [12].

III. Related Work

Constructing k-Connected k-Dominating Set (k-CDS), as the backbone of wireless networks, was first proposed in [13]. Four protocols have been proposed to construct a k-CDS, all of which are localized algorithms that depend on neighbourhood information alone. Fault-tolerant dominating set is investigated in [14]. The authors focus on k-fold dominating set in two network models. The authors in [15] focus on minimum m-connected k-tuple dominating set problem.

The concept of kmCDS has been studied in [16] and two approximation algorithms have been proposed. The authors in [17] present a centralized algorithm and a distributed deterministic algorithm to build kmCDS for general k and m. The problem of distributively constructing the k-dominating sets is examined in [18] and an incremental distributed algorithm for the same has been presented. In [3], the authors focus on fault-tolerant CDS problem in wireless networks. An approximation algorithm for kmCDS is presented.

Constructing kmCDS for general k and m has been examined in [19]. A centralized algorithm and a distributed local decision algorithm are proposed. The authors in [20] provide a Connecting Dominating Set Augmentation algorithm to form a 2-connected virtual backbone that can resist the failure of one wireless node. Constructing a connected k-dominating set (1-k-CDS) is examined in [21]. In [22], the authors propose a decentralized algorithm for constructing 2-Connected 1-Dominating Set (2-1-CDS).

2-Connected m-Dominating Set in ad hoc networks has been studied in [23]. The proposed decentralized

algorithm creates a robust backbone in ad hoc networks and is adaptable to mobile environments. In [6], the authors provide three algorithms to construct kmCDS for general k and m in wireless networks. Constructing quality fault-tolerant virtual backbone in wireless networks has been examined in [24]. The author in [25] has investigated the problem of finding small k-dominating sets in general graphs.

In [26], a greedy algorithm that constructs a minimum 1-m-CDS is presented. The authors in [27] present an integer programming formulation and an optimal algorithm to achieve routing flexibility and fault-tolerance. In [28], an approximation algorithm for minimum 3-Connected m-Dominating Set problem for m ≥ 3 using Tutte's decomposition technique has been provided.

IV. FAULT-TOLERANT IMPROVED OPTIMIZED LINK STATE ROUTING (FT-IOLSR) PROTOCOL

This section discusses kmCDS, the HELLO message format and the functioning of FT-IOLSR protocol.

A. k-Connected m-Dominating Set (kmCDS)

A CDS preserves only 1-connectivity. Because of this, many dominatees may be attached to only one dominator. Under these conditions, the failure of any one dominator results in breakage in the virtual backbone connectivity. This seriously impedes the message delivery capability of the routing protocol and ultimately degrades its performance. Whereas, in a kmCDS, the requirement of k-connectivity guarantees that, between any pair of nodes in a CDS, there exist at least k different paths. With k-connectivity, communication will not be disrupted even when up to k − 1 paths fail. k-connected virtual backbones also provide multi-path redundancy for load balancing and transmission error tolerance [6, 17, 19].

Definition 1: k-vertex connected or k-connected

A graph G is said to be k-vertex connected or k-connected if for each pair of vertices there exists at least k mutually independent paths connecting them. In other words, the graph G is still connected even after the removal of any k – 1 vertices from G [29].

The requirement of m-domination takes care of fault-tolerance and robustness of dominatee nodes which are not part of the CDS. The m-domination constraint ensures that every dominatee node has at least m neighbour dominators in a CDS. Therefore, one dominatee node can still be connected with the CDS even when up to its m − 1 dominator neighbours fail [6, 17, 19].

Definition 2: m-dominating set

An m-dominating set D_m is a set $D_m \subset V$ such that every vertex in $V \setminus D_m$ is dominated by at least m vertices in D_m [29].

Thus, with a kmCDS, every dominatee node can tolerate up to m – 1 faults on its dominators and the virtual backbone can tolerate up to k – 1 faults [26].

Definition 3: k-Connected m-Dominating Set

A set $C \subset V$ is a k-Connected m-Dominating Set (kmCDS) of graph G(V,E) if the induced subgraph G'(C,E') is k-vertex connected and the set C is also an m-dominating set of G [29].

The proposed FT-IOLSR protocol uses kmCDS to construct a fault-tolerant virtual backbone.

B. HELLO Message Format

HELLO messages, which are never forwarded, are primarily used for neighbour detection. To gather the required data about neighbours, for constructing fault-tolerant virtual backbone, fields have been added to the HELLO message format [4] of the OLSR protocol. The structure of the HELLO message used in FT-IOLSR protocol is shown in Fig. 1. The new fields that are added are Weight, Colour, NodeID, NodeConnectCount and the list of NodeConnectIDs. Weight of a node is a function used to calculate the importance of the node based on its degree, where degree refers to the number of neighbour nodes. Colour of a node could be white or black denoting that the node is a dominatee or a dominator respectively. NodeID field denotes a unique number used for identification of the node.

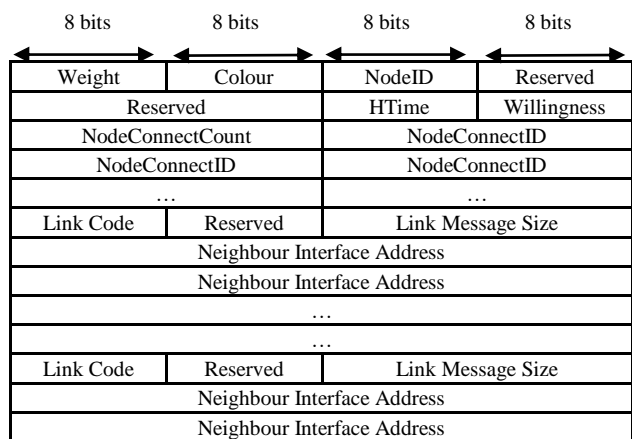| 8 bits	8 bits	8 bits	8 bits
Weight	Colour	NodeID	Reserved
Reserved		HTime	Willingness
NodeConnectCount		NodeConnectID	
NodeConnectID		NodeConnectID	
…		…	
Link Code	Reserved	Link Message Size	
Neighbour Interface Address			
Neighbour Interface Address			
…			
…			
Link Code	Reserved	Link Message Size	
Neighbour Interface Address			
Neighbour Interface Address			

Fig.1. HELLO Message Format of FT-IOLSR Protocol

After constructing CDS, each node determines whether kmCDS criterion is satisfied in its neighbourhood. If not, the node determines the list of nodes that need to be upgraded as dominators to satisfy the criterion for kmCDS. The IDs of these chosen nodes are sent to the neighbour nodes using the NodeConnectIDs field of the HELLO message. The count of this list of node IDs is sent in the NodeConnectCount field of the HELLO message. Whenever a request from neighbour nodes, for the current node to become a dominator node, is received through HELLO messages, the current node upgrades itself to become a dominator.

C. Functioning of FT-IOLSR protocol

The protocol starts with forming a Minimum Connected Dominating set (MCDS) which is 1-connected and 1-dominating, using any of the existing MCDS construction algorithms. This is followed by augmenting the MCDS to become 1-Connected m-Dominating set (1-

m-CDS), so that the virtual backbone becomes m-dominating. Next, the protocol makes the virtual backbone to be k-connected, so that the virtual backbone becomes k-connected m-dominating. As part of this process, whenever a node gets a request for becoming a dominator from other nodes, in order to satisfy k-connected m-dominating criterion, the node upgrades itself to become a dominator. Algorithm 1 depicts the overall functioning of the FT-IOLSR protocol.

Algorithm 1. FT-IOLSR_CDS()

1. Construct the MCDS virtual backbone
2. Make the virtual backbone to be 1-connected m-dominating using algorithm FT-IOLSR_mCDS()
3. Augment the CDS in step 2 to be k-connected using algorithm FT-IOLSR_kmCDS()
4. Nodes become dominators using algorithm FT-IOLSR_kmCDSUpdation(), on receiving request from neighbour nodes through HELLO messages

---

(i) 1-Connected m-dominating virtual backbone

Each dominatee node determines its number of 1-hop neighbour nodes and out of these, the number of dominator nodes is found out. The node then verifies whether it is connected to m number of dominator nodes to satisfy the m-dominating criterion. If the m-dominating criterion is satisfied, no further action on the part of the node is necessary.

On the other hand, if the node is connected to less than m number of dominators, additional nodes need to be made dominators. For this, the node selects the required number of neighbour dominatee nodes, with respect to the node's weight in descending order. Weight of a node is calculated by finding the degree of a node, which refers to the number of its neighbour nodes. If the required number of nodes are unavailable, the available dominatee neighbour nodes are selected. The selected neighbour nodes are then requested to become dominators. The node sends this request in the next outgoing HELLO message. Algorithm 2 depicts the m-dominating process.

Algorithm 2. FT-IOLSR_mCDS()

1. $N(u)$ is the set of 1-hop neighbours reachable by a dominatee node u
2. t = number of nodes in $\{N(u)\}$ and n = number of nodes in $\{N(u) \cap MCDS\}$
3. m denotes the m-domination parameter
4. if ($n >= m$) then
5.   1-Connected m-Dominating criterion is satisfied
6. end if
7. if ($n < m$) then
8.   Find nr, the number of dominatee neighbournodes
9.   r represents the shortage in m-domination
10.   if ($nr <= r$) then
11.   Request the remaining nodes in the set $\{N(u) - \{N(u) \cap MCDS\}\}$ to become dominators
12.   else
13.   Request the set of top r neighbour nodes in the set $\{N(u) - \{N(u) \cap MCDS\}\}$, with respect to weight in descending order, to become dominators
14.   end if
15. end if

(ii) k-connected m-dominating virtual backbone

To make the virtual backbone k-connected, each dominator node finds the number of mutually independent paths between itself and each of its neighbour dominator nodes. It then determines whether the available number of paths to each of its neighbour dominator nodes is greater than or equal to k. If so, the k-connected criterion is satisfied and no change to the virtual backbone needs to be made by the dominator node. For the neighbour dominator nodes, for which this criterion is not satisfied, the dominator node determines the other viable paths through which the k-connected criterion can be satisfied. These paths should be mutually independent of the already available paths connecting the node under consideration with a given neighbour dominator node. Also, the paths with shortest length should be given preference.

If the available paths are less than the number of paths required to satisfy the k-connected criterion, then all the paths are chosen. If the available paths are more than the required number of paths to satisfy the k-connected criterion, the required number of paths are chosen among them, taking shortest path as the selection criterion. The dominatee nodes on these newly determined paths are requested to become dominators in the next outgoing HELLO message. The process of making the virtual backbone k-connected is depicted in Algorithm 3.

Algorithm 3. FT-IOLSR_kmCDS()

1. u is the current dominator node and k denotes the k-connectivity parameter
2. $N_{dom}(u)$ is the set of neighbour dominator nodes reachable from u
3. Compute mutually independent paths $P_{dom}(i)$ between u and i passing through only dominator nodes, where $i \in N_{dom}(u)$
4. $n_{dom}(i)$ = number of paths in $P_{dom}(i)$
5. if ($n_{dom}(i) >= k$) then
6.   k-Connected m-Dominating criterion is satisfied
7. else
8. $x_{dom}(i)$ denotes the set of nodes used to find $P_{dom}(i)$
9.   Determine all the mutually independent paths $P_{newdom}(i)$ between node u and i through nodes excluding the ones in $x_{dom}(i)$, giving preference to shortest paths
10. $n_{newdom}(i)$ = number of paths in $P_{newdom}(i)$
11. $k_{rem}(i) = k - n_{dom}(i)$
12.   if ($n_{newdom}(i) <= k_{rem}(i)$) then
13.   Use all the paths in $P_{newdom}(i)$
14.   Find the largest subset of nodes w in $P_{newdom}(i)$ such that $\{w \cap CDS\} = \{\}$ and request the nodes in w to become dominators
15.   else
16.   Use top $k_{rem}(i)$ paths from $P_{newdom}(i)$ with respect to the path length in increasing order, denoted by $P_{selnewdom}(i)$
17.   Find the largest subset of nodes w in $P_{selnewdom}(i)$ such that $\{w \cap CDS\} = \{\}$ and request the nodes in w to become dominators
18.   end if
19. end if

(iii) kmCDS updation

A node may be requested by the neighbour nodes to become a dominator to form kmCDS. These requests are

transmitted by the neighbour nodes using HELLO messages. Whenever a node receives a HELLO message, the list of IDs of the nodes which have been requested to become dominators is verified to determine whether the node's ID is part of it. If so and if the current node's ID is less than the node ID of the sender of the HELLO message, the node becomes a dominator, provided it is a dominatee at that point of time. If not, no action needs to be taken on the part of the node. The steps of the updation process are given in Algorithm 4.

Algorithm 4. FT-IOLSR_kmCDSUpdation()
1.   u denotes the current node
2.   x is the originator of the HELLO message
3.   C is the set of dominator nodes
4.   v denotes the set of nodes received through HELLO message in NodeConnectID field for becoming dominators
5.   if $(u \subset v)$ and $(u_{id} < x_{id})$ then
6. $C = C \cup \{u\}$
7.   else
8.   Discard the received list of nodes
9.   end if

V. Simulation

The simulation environment and the simulation results are presented in this section.

A. Simulation Environment

Using the NS-3 simulator [30], the performance of FT-IOLSR and OLSR protocols is assessed. Simulations have been carried out for a network consisting of 50 nodes which are spread over an area of 1250m x 1250m. Random waypoint mobility model is used with node speed varied as 5, 10 and 15 m/s. The pause time has been set as 2 seconds and 5 seconds. Simulation is run for a total of 500 seconds.

FT-IOLSR protocol and OLSR protocol have been assessed by studying the impact of mobility on packet delivery ratio (PDR), with various node speed and pause time combinations. The performance of the FT-IOLSR protocol is also assessed under varying node speed keeping pause time constant and also under varying pause time keeping the node speed constant. Also, the effect of the network size on the performance of both the protocols is examined. Further, the impact of connectivity factor

and the impact of domination factor on the performance of FT-IOLSR protocol is examined.

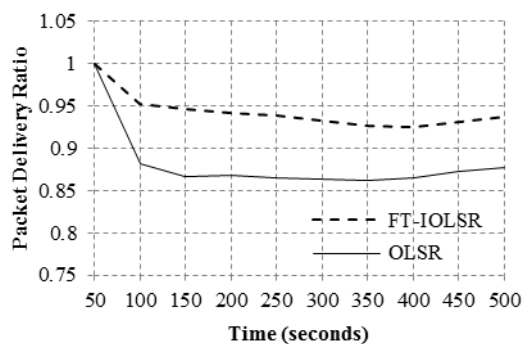### B. Simulation Results

(i) Impact of mobility

Fig. 2(a) to 2(f) reflect the PDR of FT-IOLSR and OLSR protocols for various combinations of node speed and pause time. The node speed is varied as 5, 10 and 15 m/s and the pause time is set as 2 seconds and 5 seconds.

It is clearly evident from the figures that, for different node speed and pause time combinations, PDR of FT-IOLSR protocol is substantially higher than that of the OLSR protocol. This is because, due to mobility, nodes get detached from one part of the network and attach themselves to another part of the network. Because of this, the already available routes to different destinations may become obsolete and the packets do not reach their destinations, causing packet loss. The movement of the nodes belonging to the MPR set in the OLSR protocol affects its performance, as non-MPR set nodes may not be in the vicinity of any node belonging to the MPR set. Also the connectivity among the nodes of the MPR set may be lost due to node mobility.
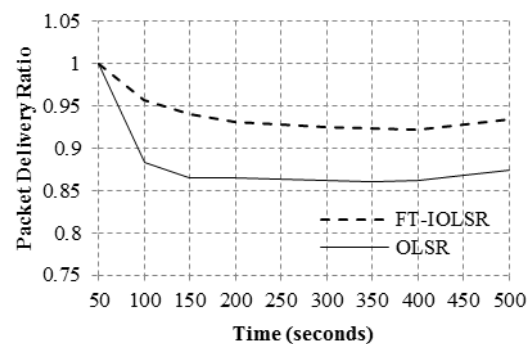
The FT-IOLSR protocol provides a solution to this by employing fault-tolerant CDS where the virtual backbone is k-connected m-dominating. This, as already elaborated, provides fault-tolerance up to k – 1 path failures and m – 1 dominator failures. Because of this, the k-connected m-dominating virtual backbone formed by the FT-IOLSR protocol is able to withstand disconnections owing to node mobility. As a result, the packet delivery is better in FT-IOLSR protocol and PDR is comparatively higher than that of the OLSR protocol.

(a) Impact of varying node speed and constant pause time

The impact of varying node speed, keeping the pause time constant, on the performance of the FT-IOLSR protocol is next examined. Fig. 3(a) and 3(b) show the PDR of FT-IOLSR with varying node speed keeping the pause time constant as 2 seconds and 5 seconds respectively. The node speed is varied as 5 m/s, 10 m/s and 15 m/s.

(a) Speed = 5, Pause = 2

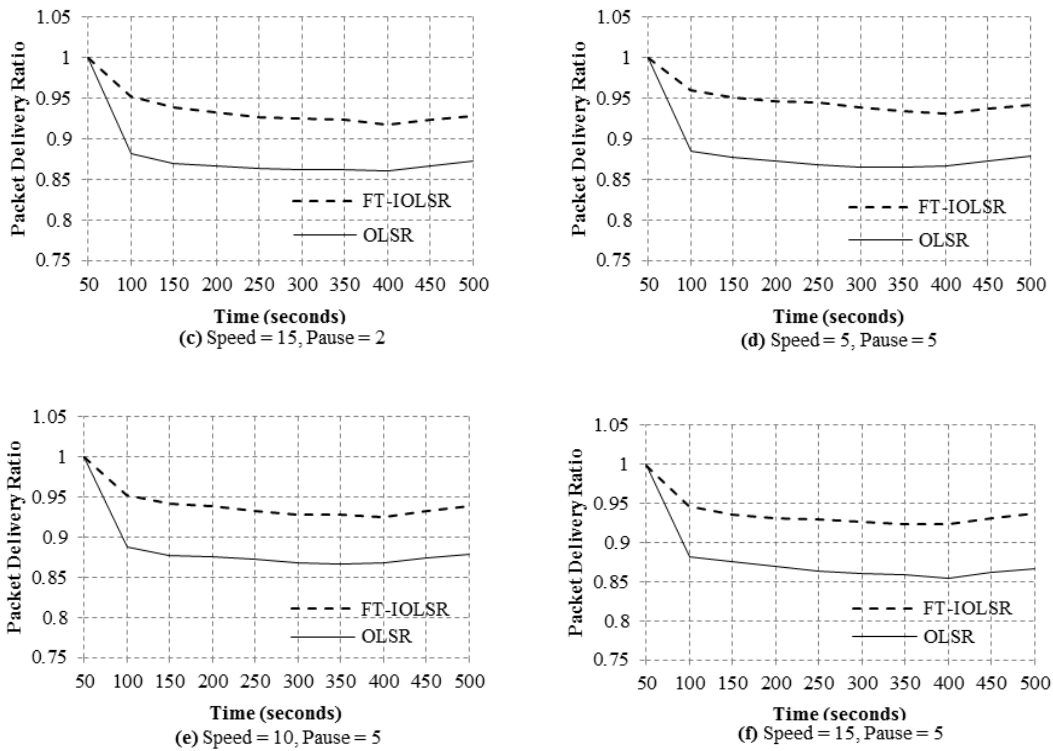**(b)** Speed = 10, Pause = 2

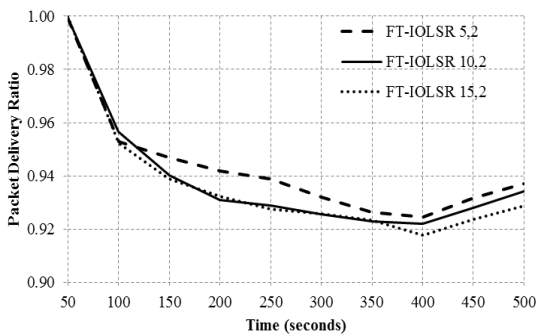Fig.2. PDR of FT-IOLSR and OLSR Protocol

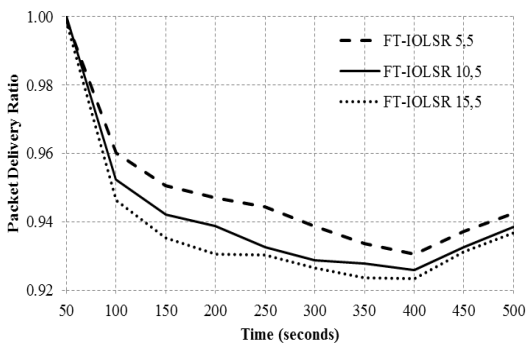Fig.3(a). Impact of Varying Node Speed with 2 Seconds Pause Time

Fig.3(b). Impact of Varying Node Speed with 5 Seconds Pause Time

FT-IOLSR protocol provides better PDR at node speed of 5 m/s as compared to 10 m/s and 15 m/s. This is observed in both the cases of 2 seconds and 5 seconds pause time. There is gradual decline in PDR as the node speed is increased to 10 m/s and 15 m/s. At lower speed, there are fewer disconnections owing to mobility, and hence improved PDR is observed. It is evident that the PDR gradually decreases with increase in speed in both the cases of pause time under consideration.

(b) Impact of varying pause time and constant node speed

The performance of the FT-IOLSR protocol is examined with respect to varying pause time keeping the node speed as constant. The results of the PDR are as shown in Fig. 4(a) to 4(c), where the node speed is 5 m/s, 10 m/s and 15 m/s respectively, with pause time being set as 2 seconds and 5 seconds. The results in these figures indicate that there is an increase in PDR when the pause time is increased, for node speed of 5 m/s and 10 m/s. However, as the node speed increases to 15 m/s, the improvement is very minimal. It can be inferred that at lower node speeds, the increase in pause time has a positive effect on the PDR.
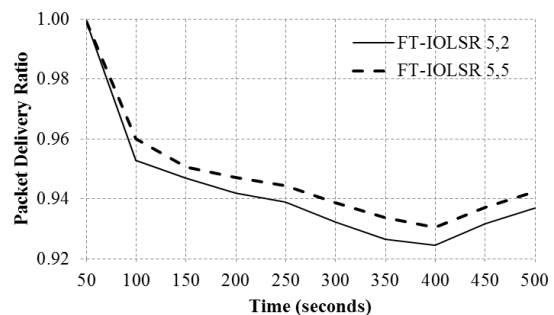

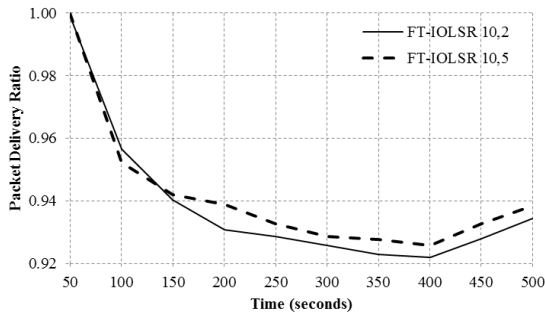Fig.4(a). Impact of Varying Pause Time with 5 m/s Node Speed

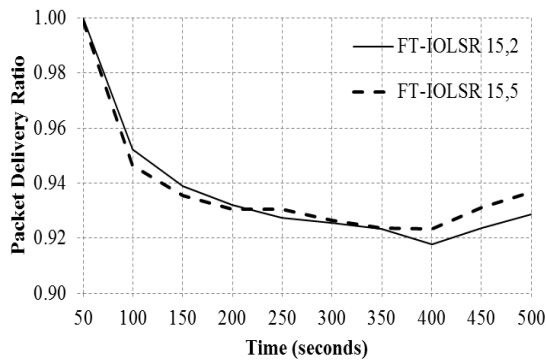Fig.4(b). Impact of Varying Pause Time with 10 m/s Node Speed

Fig.4(c). Impact of Varying Pause Time with 15 m/s Node Speed

A comparative chart illustrating the performance of the FT-IOLSR protocol in terms of PDR for various combinations of node speed and pause time is shown in Fig. 5. From the figure, it is inferred that, as the node speed increases, the PDR decreases, whereas with increase in pause time the PDR increases.
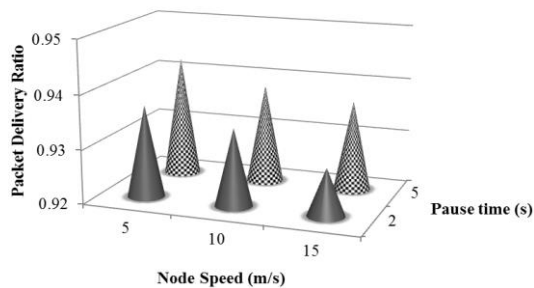

Fig.5. PDR of FT-IOLSR Protocol for Various Node Speed and Pause Time Combinations

(ii) Impact of network size

The effect of network size on the PDR achieved by FT-IOLSR protocol and OLSR protocol is examined. The results are shown in Fig. 6, where the network size is varied as 25, 50, 75 and 100 nodes. It is observed that, as the number of nodes in the network increases, the PDR of both the protocols increases. There is comparatively substantial improvement in PDR in the case of 25 and 50 nodes in both FT-IOLSR and OLSR protocols. Then, for

the other two increments of 25 nodes, there is gradual increase in PDR.

The increase in PDR can be attributed to the increase in node density, as a result of increase in network size. Increased node density leads to more connectivity in the network and hence, better packet delivery can be achieved with increase in network size. Further, it is evident that FT-IOLSR protocol consistently performs better than the OLSR protocol due to its fault-tolerance capability.

(iii) Impact of connectivity factor

Next the performance of the FT-IOLSR protocol is evaluated under varying connectivity factor and the results are shown in Fig. 7. Connectivity factor is described by the k value which is varied from 1 to 4 and the dominating factor is set as 2.
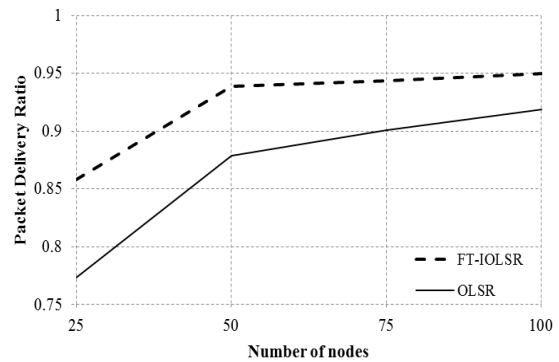

Fig.6. Impact of Network Size

It is observed that as the connectivity factor increases from 1 to 3, there is gradual improvement in the PDR. But, as the connectivity factor gets increased to 4, substantial improvement in PDR is observed. It is therefore inferred that with increase in connectivity factor, the protocol provides better PDR.

As the connectivity factor increases, more mutually independent paths are available between any two dominators and the network is therefore able to withstand multiple dominator failures. This results in successful delivery of more number of packets. Hence, virtual backbones with higher connectivity factor perform better as compared to those with lower k-connectivity values.
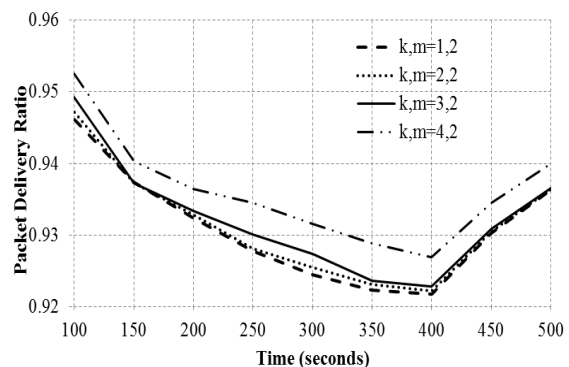

Fig.7. Impact of Connectivity Factor on FT-IOLSR Protocol

(iv) Impact of domination factor

PDR achieved by the FT-IOLSR protocol for varying domination factor is shown in Fig. 8. Connectivity factor k is set as 2, whereas the domination factor m is varied from 1 to 4. The results indicate that increase in domination factor leads to improved PDR. When the domination factor is increased, the number of dominators to which a dominatee is connected increases and therefore, a dominatee node is able to survive multiple dominator failures. This in turn, substantially improves the fault-tolerance capability of the network and hence, an improvement in PDR is observed.
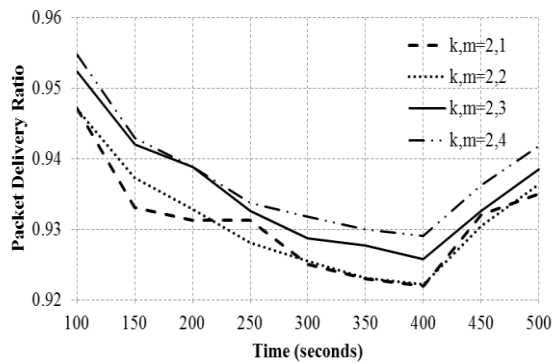

Fig.8. Impact of Domination Factor on FT-IOLSR Protocol

VI. CONCLUSION

The proposed FT-IOLSR protocol used kmCDS to construct the fault-tolerant virtual backbone that can withstand up to m − 1 dominator node failures and up to k − 1 path failures. It effectively replaced the MPR set used in the OLSR protocol. Simulations were carried out comparing the performance of the FT-IOLSR protocol against OLSR protocol. The performance of the FT-IOLSR protocol has been assessed in terms of PDR under varying mobility, network size, domination factor and connectivity factor. The simulation results indicate that the FT-IOLSR protocol performed better under various combinations of node speed and pause time and varying network size. Further, with increase in domination factor and connectivity factor, the protocol exhibited improved performance.

REFERENCES

[1] 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, 75(1), 56-73.
[2] Kim, D., Gao, X., Zou, F., & Wu, W. (2011). Construction of fault-tolerant virtual backbones in wireless networks. In Y. Xiao, F. H. Li & H. Chen (Eds.), Handbook of Security and Networks (pp. 465-486). Singapore: World Scientific.
[3] Zhang, N., Shin, I., Zou, F., Wu, W., & Thai, M. T. (2008). Trade-off scheme for fault tolerant connected dominating sets on size and diameter. Proceedings of the 1st ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC'08). Hong Kong SAR, China (pp. 1-8).
[4] Clausen, T. H., & Jacquet. P. (2003, October). Optimized link state routing protocol (OLSR), Request for Comments: 3626. Retrieved from https://www.ietf.org/rfc/rfc3626.txt.
[5] Wu, J. (2002). Dominating-set-based routing in ad hoc wireless networks. In I. Stojmenovic ´(Ed.), Handbook of Wireless Networks and Mobile Computing (pp. 425-450). New York: Wiley.
[6] Li, Y., Wu, Y., Ai, C., & Beyah, R. (2012). On the construction of k-connected m-dominating sets in wireless networks. Journal of Combinatorial Optimization, 23(1), 118-139.
[7] Kim, D., Wu, Y., Li, Y., Zou, F., & Du, D. (2009). Constructing minimum connected dominating sets with bounded diameters in wireless networks. IEEE Transactions on Parallel and Distributed Systems, 20(2), 147-157.
[8] 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, 12(4), 417-428.
[9] 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).
[10] 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).
[11] Gao, B., Ma, H., & Yang, Y. (2005). A new distributed approximation algorithm for constructing minimum connected dominating set in wireless ad hoc networks. In J. Cao, L. T. Yang, M. Guo, & F. Lau (Eds.), Parallel and Distributed Processing and Applications (pp. 352–356). Berlin Heidelberg: Springer.
[12] Li, Y., Zhu, S., Thai, M., & Du, D. (2004). Localized construction of connected dominating set in wireless networks. Proceedings of US National Science Foundation International workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor and Peer-to-Peer Networks (TAWN'04). Chicago, USA.
[13] Dai, F., & Wu, J. (2006). On constructing k-connected k-dominating set in wireless ad hoc and sensor networks. Journal of Parallel and Distributed Computing, 66(7), 947-958.
[14] Kuhn, F., Moscibroda, T., & Wattenhofer, R. (2006). Fault-tolerant clustering in ad hoc and sensor networks. Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS '06). Lisbon, Portugal.
[15] Shang, W., Wan, P., Yao, F., & Hu, X. (2007). Algorithms for minimum m-connected k-tuple dominating set problem. Theoretical Computer Science, 381(1–3), 241-247.
[16] Thai, M. T., Zhang, N., Tiwari, R., & Xu, X. (2007). On approximation algorithms of k-connected m-dominating sets in disk graphs. Theoretical Computer Science, 385(1–3), 49-59.
[17] Wu, Y., Wang, F., Thai, M. T., & Li, Y. (2007). Constructing k-connected m-dominating sets in wireless sensor networks. Military Communications Conference (MILCOM 2007). Orlando, FL, USA (pp. 1-7).
[18] Couture, M., Barbeau, M., Bose, P., & Kranakis, E. (2008). Incremental construction of k-dominating sets in

wireless sensor networks. Ad Hoc & Sensor Wireless Networks, 5(1-2), 47-67.

[19] Wu, Y., & Li, Y. (2008). Construction algorithms for k-Connected m-Dominating sets in wireless sensor networks. Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08). Hong Kong SAR, China (pp. 83-90).

[20] Wang, F., Thai, M. T., & Du, D. (2009). On the construction of 2-connected virtual backbone in wireless networks. IEEE Transactions on Wireless Communications, 8(3), 1230-1237.

[21] Bian, Y., Yu, H., & Zeng, P. (2009). Construction of a fault tolerance connected dominating set in wireless sensor network. International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '09). Zhangjiajie, Hunan (pp. 610-614).

[22] Schleich, J., Bouvry, P., & An, L. T. H. (2009). Decentralized fault-tolerant connected dominating set algorithm for mobile ad hoc networks. Proceedings of the 2009 International Conference on Wireless Networks (ICWN 2009). Las Vegas Nevada, USA (pp. 354-360).

[23] Schleich, J., Danoy, G., Bouvry, P., & An, L. T. H. (2009). Blackbone2, an efficient deterministic algorithm for creating 2-connected m-dominating set-based backbones in ad hoc networks. Proceedings of the Seventh ACM International Workshop on Mobility Management & Wireless Access, (MobiWac'09). Tenerife, Canary Islands, Spain (pp. 91-98).

[24] Wang, W., Kim, D., An, M. K., Gao, W., Li, X., Zhang, Z., & Wu, W. (2013). On construction of quality fault-tolerant virtual backbone in wireless networks. IEEE/ACM Transactions on Networking, 21(5), 1499-1510.

[25] Foerster, K. (2013). Approximating fault-tolerant domination in general graphs. Proceedings of the Tenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO). SIAM (pp. 25–32).

[26] Zhou, J., Zhang, Z., Wu, W., & Xing, K. (2014). A greedy algorithm for the fault-tolerant connected dominating set in a general graph. Journal of Combinatorial Optimization, 28(1), 310-319.

[27] Ahn, N., & Park, S. (2015). An optimization algorithm for the minimum k-connected m-dominating set problem in wireless sensor networks. Wireless Networks, 21(3), 783-792.

[28] Wang, W., Liu, B., Kim, D., Li, D., Wang, J., & Jiang, Y. (2015). A better constant approximation for minimum 3-connected m-dominating set problem in unit disk graph using Tutte decomposition. Proceedings of the 34th IEEE International Conference on Computer Communications (INFOCOM 2015). Hong Kong (pp. 1796-1804).

[29] Wu, Y., & Li, Y. (2009). Connected Dominating Sets. In H. Liu, X. Chu, & Y. Leung (Eds.), Ad Hoc and Sensor Wireless Networks: Architectures, Algorithms and Protocols (pp. 19-39). Bentham Science Publishers.

[30] https://www.nsnam.org

Authors' Profiles

Vinayagam P. S. is working as an Assistant Professor in the Department of Computer Science, Pondicherry University Community College. He has 15 years of teaching experience. His research interests include algorithms and networks.