

Original Research Paper

A Novel Proactive Routing Method for Mobile Ad Hoc Networks

Bambrik Ilyas and Didi Fedoua

*Department of Computer Science, Laboratory of Research in Informatics of Tlemcen (LRIT),
Abou Bekr Belkaid University, Tlemcen, Algeria*

Article history

Received: 09-08-2014

Revised: 10-05-2015

Accepted: 12-07-2015

Corresponding Author:

Bambrik Ilyas

Department of Computer
Science, Laboratory of

Research in Informatics of

Tlemcen (LRIT), Abou Bekr

Belkaid University, Tlemcen,
Algeria

Email: ilyas9111@yahoo.fr

Abstract: In recent years, the Mobile Ad hoc Networks (MANETs) became more popular due to their multiple advantages such as the low setup cost, auto-configuration and auto-healing features. This kind of network is distinguished by its dynamic and distributed nature since it does not contain any form of centralized infrastructure. These characteristics have led to the emergence of new routing protocols specifically designed for dynamic topologies. Some protocols construct and maintain paths before they are needed (proactive protocols). However, this method generates a massive volume of overhead. Other protocols reduce the overhead by constructing paths only when it is necessary (reactive protocols). On the other hand, this method is not well adapted when the topological changes are frequent and can increase the transmission delay. Though the features presented by this type of network seem appealing, maintaining the network connectivity in a distributed fashion is a complicated task and could drain a significant piece of the network resources. Moreover, considering that the network members are usually embedded devices equipped with limited resources, it is vital to minimize the resources spent by the routing protocol. In this study we compare the various routing strategies designed for MANET and propose a proactive routing algorithm that consumes an acceptable quantity of the network members resources. Furthermore, we present how our protocol works and multiple tests to show the protocol reactions to the different topological parameters.

Keywords: Routing Protocol, MANET, Proactive, Reactive, Overhead, Network Resources

Introduction

With the recent technological advances in the field of telecommunication, a new type of wireless network called Mobile Ad hoc Network (MANET) has emerged. Unlike the traditional wireless networks, MANETs are formed by mobile hosts (smart phones, laptops, ...etc) solely. Instead of relying on a preset infrastructure, all the network members collaborate in a distributed fashion in order to form multi-hop paths. Furthermore, the network members are able to move freely which makes the network topology highly flexible. Due to the interesting features of the MANET, this network technology has become a core feature in other networks such as the Wireless Mesh Network (WMN) and the Vehicular Ad hoc Network (VANET). The ability of operating without a centralized infrastructure makes this type of network very suitable for situations where

deploying an infrastructure is difficult or unwanted and more importantly, this technology can be viewed as an ideal platform to support the pervasive systems.

Since the topology can change randomly, the traditional routing solutions cannot function in this type of network. Therefore, a new type of routing protocol appeared that can operate in a dynamic topology. Based on the manner with which the paths between the network members are created, three categories of MANET routing protocols exist. First, the proactive protocols create and maintain the paths to all the potential destinations in the network. Accordingly, when a network member requires a path to a given destination, the data transfer starts without delay. To apply this strategy, the network members periodically broadcast their topological information and process the incoming routing messages. As a consequence, this type of

protocol can consume a portion of the mobile hosts resources (bandwidth, energy and computation time).

Secondly, the reactive protocols create the paths on demand instead of building a path to every possible destination. In this strategy, a path between a pair of nodes is created and maintained as long as it is used. When a path toward a given destination is needed, a request/wait for reply process is launched. Then, the path requests are rebroadcast by the intermediate nodes until the destination is found. Afterwards, a reply is sent to the source through the found route. Thus, unlike the proactive protocols, this category of protocol lowers the overhead when the transmission requests are uncommon.

Although this routing approach preserves the resources of the network, the reactive path building process can introduce a transmission delay. Inevitably, this problem expands when the topology changes frequently. Furthermore, this strategy is ineffective when the number of transmission demands is elevated and can consume as much resources as the proactive method in such a situation. Logically, when the user application requires a constant connectivity between the network members, it is more appropriate to apply the proactive approach than to create a path to each destination separately. Moreover, whereas the proactive approach automatically optimizes the available paths, the reactive protocol first informs the source about the path rupture so that a new search for the destination can be launched.

The mixture between the proactive and the reactive method created a new type of protocol called hybrid. First, the proactive method is applied to collect the routing information about the close neighborhood (3 or 4 hops away). After that, in case a path to a remote destination is needed, the reactive approach is launched. Consequently, this kind of protocol generates less overhead than the proactive protocols and reduces the transmission delay by comparison with the reactive protocols. As a substitute to the reactive path building process, the request messages are forwarded through routing zones until the destination is found instead of overflowing the entire network. Nonetheless, this method does not produce necessarily the shortest paths because, the search for the destination, is zone driven.

Some protocols organize the network in a hierarchical manner to attribute a role for each network member. The idea is to represent the network in a form of multiple clusters where each cluster has one representative responsible of routing functions, labeled as a Cluster Head. Mobile hosts that belong to more than one cluster are used for communication between clusters and are considered as Gateways. As a result, the hierarchical organization minimizes the routing operations performed by the cluster members since the routing functions are insured only by the Cluster Heads and Gateways. The downside is that the concentration of data flows on the Cluster Heads and Gateways can lead to network congestion and eventually, node failure.

Unquestionably, the routing protocol is the kernel function behind all the advantages of the MANET. Considering the differences in the strategies applied by the routing protocols, the performances of the protocol depend on the context for which the MANET is used. Hence, multiple extensions were proposed in order to increase the adaptability of the routing protocols in different situations. Some researchers focused their work on how to balance the traffic load in order to increase the network life time (Day *et al.*, 2011). Other studies were oriented toward Quality of Service (QoS) awareness (Poonkuzhali *et al.*, 2014) and security insurance (Gopalakrishnan and Ganeshkumar, 2015; Chandramohan and Kamalakkannan, 2014).

For instance, in a scenario where the transmission requests are rare, the reactive and hybrid protocols can outperform the proactive protocols. On the other hand, in a situation where the user application requires a constant connectivity between the users, the proactive protocols are more suitable. In addition, the proactive method offers many advantages such as its better awareness of the topological events and low transmission delay. However, the complexity of the proactive protocols make their performances depend on the topological parameters. Besides, the MANET components are usually equipped with a restricted CPU power, limited bandwidth and energy. For that reason, the constant advertisement/processing of the routing information can accelerate the exhaustion of the mobile hosts energy; especially in a large networks.

In order to lower the overhead of the proactive strategy, many methods have been proposed. In this study we present a proactive algorithm based on a new routing information exchange/processing method that aims to reduce the complexity of the routing function. Although the proposed algorithm is proactive, the resulting overhead is fairly reasonable by comparison with other protocols from the same category. Furthermore, we propose several series of simulations to test the proposed solution resiliency based on different topological parameters (nodes speed of movement, network size and density).

This paper is organized as the following: In section I we discuss the different routing algorithms proposed for MANET. In section II, we provide the core algorithm for our protocol. Then, in section III we explain how the protocol is tested. In section IV, we discuss the results obtained from the protocol tests. Finally, section V concludes this paper.

Literature Study

The Proactive Protocols

Due to the distributed and unpredictable nature of the MANET, numerous routing protocols have been

designed to execute the routing function in dynamic topologies. Destination Sequenced Distance Vector (DSDV) developed by Perkins and Bhagwat (1994) is a proactive protocol that chooses the paths based on how fresh they are. To determine the freshness of the paths, each path is tagged with the corresponding destination sequence number so that the node can select the most recent path. In addition, important topological changes are immediately announced by the network members. Consequently, this protocol efficiently eliminates routing loops and accelerates the convergence. However, when the topology changes frequently, the volume of routing messages produced by this protocol, expands.

Optimized Link State Routing (OLSR), designed by Clausen and Jacquet (2003) is a proactive protocol based on the Link State algorithm. This protocol optimizes the dissemination of the control messages by selecting for each network member, a subset of its neighborhood called Multipoint Relays (MPR). Simply, this subset contains the minimum set of neighbors that can enable a network member to reach all the adjacent nodes two hops away, or less. Afterwards, all the neighbors process the control messages but only the selected MPRs forward them. Consequently, the links that are disseminated in the network, are limited.

The strength of this protocol is its adaptability to dense topologies. Instead of announcing all the links with the neighbors, only the necessary links are broadcast. Although the MPR selection process requires a periodic treatment of the neighbors set, this mechanism eliminates the asymmetric link problem. Also, from a security point of view, relaying only on the selected MPRs eliminates the misbehaving host problem since, the network member that does not share any routing information, is excluded from the MPR list.

The idea applied by Gerla *et al.* (2002) in Fisheye State Routing (FSR), is to use the Fisheye algorithm which was initially designed for the compression of visual data. When applied on routing information, the network members advertise the description of their near neighborhood more frequently than the description of the remote destinations. Thus, the outcome of this method is the reduction of the control message size and a slight inaccuracy while building paths to the remote destinations. However, since the routing decisions are made hop by hop, the routing imprecision can be insignificant because the information about the destination is more accurate for the network members relatively closer to it. Although this method slows down the convergence of the protocol, this strategy is well adapted to large topologies. Furthermore, Andreas *et al.* (2009) proposed an integration of this method in OLSR to lower the overhead. Figure 1 is an example of how FSR organizes the routing information.

Neumann *et al.* (2008) proposed a protocol called Better Approach to Mobile Ad hoc Networking

(BATMAN) which incorporates a new method for path building. This protocol generates control messages called Originator Messages (OGMs) and upon receipt, the distance to each destination is incremented and the message is rebroadcast. Subsequently, the paths are selected based on the number of OGMs received. The advantage of this protocol is that it produces paths with high throughput and low packet loss. On the other hand, the OGMs are advertised more frequently than the control messages of other proactive protocols and disseminating them in the network increases the overhead.

Distance Routing Effect Algorithm for Mobility (DREAM), created by Basagni *et al.* (1998), is a proactive protocol similar to FSR. This protocol collects the geographical descriptions of the network members through a localization system and uses them to construct paths. Two mechanisms compose this protocol: (a) the distance effect, (b) the node relative speed. The idea of the first mechanism is that the more a node is distant, the less it appears to be moving. Therefore, this protocol advertises the routing information representing the close nodes more often than the description of the destinations faraway. The difference between this method and FSR is that DREAM measures the geographical distance between the nodes instead of hop count.

In the second mechanism, the nodes moving at a high speed advertise their routing information more often than those moving slowly. Furthermore, this protocol considers that the next hop toward the destination is the closest one to it according to the collected geographical information. However, this perception might not be always true. Hence, making this protocol inaccurate in some scenarios. Besides, the correlation between the broadcast frequency of the control messages and the network member speed of movement, increases the overhead when the network members are moving fast.

Typically, the routing protocols for MANET tend to build the shortest path. Juliusz (2010) created a new proactive protocol, called Babel, based on the ETX metric proposed by De Couto *et al.* (2003). Basically, this protocol calculates the probability of data transmission over a link without packet loss. To achieve this, the number of control messages received from a neighbor and the expected number of control messages over a period of time, are compared. Afterwards, the neighbors proposing the highest probabilities of transmission without packet loss, are chosen for path construction purposes. As a result, this protocol promotes the paths that have a low signal interference. Additionally, the sequence number idea of DSDV, is applied in this protocol. Nonetheless, the problem of this protocol is that the computation of the ETX metric requires a recurrent broadcast of the control messages. Consequently, the routing overhead of this protocol is elevated.

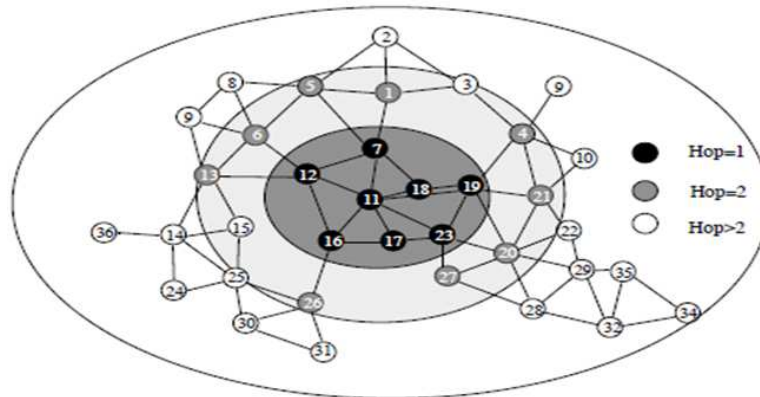


Fig. 1. Fisheye state routing (Gerla *et al.*, 2002)

The Reactive Protocols

Whereas the proactive protocols main focus is to maintain the paths between all the network members, reactive protocols create a path on demand only. Designed by Perkins *et al.* (2003), Ad hoc On demand Distance Vector (AODV) is a reactive protocol that applies a request/wait for reply cycle to construct a path. When a network member needs a path to a given destination, the search is initiated by the broadcast of a Route Request (RREQ) message. After receiving a RREQ, the intermediate node creates a reverse path to the source and rebroadcast the request only once. Afterwards, when the destination is reached or a node in possession of a recent path (by comparison with the sequence number included in the RREQ) is attained, a Route Reply (RREP) is forwarded toward the source to establish the transmission path.

The main disadvantage of AODV is its inadaptability to the frequent topological changes. When the network members movement affects the transmission path, first, the source is informed about the path rupture. Then, the search for the destination is restarted from the beginning. To limit the transmission delay problem, Alshanyour and Baroudi (2008) proposed an upgraded version of this protocol by incorporating a local route repair method. This mechanism can reduce the transmission delay due to path rupture especially when the network members are moving slowly or at a medium speed. Furthermore, in the work of Surjeet *et al.* (2014) and Kanniche *et al.* (2011), two bandwidth estimation strategies based on AODV, are proposed in order to provide the QoS support for the data streams.

Designed by Johnson *et al.* (2007), Dynamic Source Routing (DSR) applies a similar path construction process to the one applied in AODV. The difference is that the intermediate network members include their IDs in the RREQ instead of memorizing the reverse path. Moreover, when the path is sent back to the source, the

found path is included in all the data packets. As a result, this protocol requires less storage capacity and a rupture in the transmission path is immediately detected which eliminates routing loops. However, including the path in all the RREQ/RREP and data messages generates a considerable quantity of routing overhead.

The reactive protocols developed by Marina and Das (2006) and Park and Corson (1997) create multiple paths toward the same destination. This strategy can improve the protocol adaptability to frequent topological events. Instead of maintaining only one path toward the destination, Ad hoc on demand Multipath Distance Vector (AOMDV) and Temporally Ordered Routing Algorithm (TORA) maintain all the found paths during the search for the destination. After that, in case the primary transmission path is broken, one of the alternative paths is used by the source. Unless all the alternative paths are broken, this approach can significantly shorten the transmission delay. On the other hand, creating and maintaining multiple paths to the same destination, increases the overhead. In order to introduce the QoS awareness in this type of protocol, Sensarma and Majumder (2013) proposed an extension of TORA based on the Ant colony method (Dorigo and Gambardella, 1997).

The Associativity Based Routing (ABR) protocol, developed by Toh (1997), is a reactive protocol that promotes the path that has a high associativity degree. To run this protocol, each member of the network attributes an associativity degree to its neighbors by counting the number of the HELLO messages received. During the search for the destination, the associativity degree of the explored links are included in the request messages. Subsequently, when the destination is reached, the path with the highest associativity degree is selected. Also, this protocol incorporates a local path repair mechanism to reduce the transmission delay in case of a path failure. Although the local path repair is effective when the network members are moving at a

medium/low pace, this method increases the transmission delay when it fails.

Signal Stability-Based Adaptive Routing (SSA), presented in the work of Dube *et al.* (1997), is a reactive protocol that selects the paths based on the signal strength between the neighbors. First, for each neighbor, the link signal quality is recorded as “high” or “low”. Then, when the search for the destination is launched, only the path requests received from the “high” signal strength links, are forwarded. As a result, this protocol creates paths formed by “high” quality links. The disadvantage of this method is that, if the first attempt to build a path with “high” quality links fails, the transmission source then restarts the search and the “low” quality links are accepted during the second attempt. Thus, the transmission delay is increased.

Ko and Vaidya (2000) proposed another protocol called Location Aided Routing (LAR) that collects the geographical coordinates of the mobile hosts. This reactive protocol explores the existing geographical information about the destination (last known location, time elapsed since the last communication, speed and direction of movement) to narrow down the location where the destination might be found. First, the source calculates the relative distance that separates it from the destination and includes it in the request message. Subsequently, the intermediate nodes receiving the request message will rebroadcast it only if they are closer to the destination. In order to enlarge the search area, the distance separating the destination from the source, is increased at the beginning of the process. Consequently, the probability of finding the destination, is increased. However, supposing that the next hop leading to the destination is the closest one to it (geographically), can be false. As a result, the path repair mechanism can falsely fail which leads to restarting the search for the destination and including all the network members in the process. Hence, if the path repair fails, the transmission delay is increased.

The Hybrid and Hierarchical Protocols

Hybrid protocols usually create multiple routing zones by applying the proactive approach. Then, the reactive approach is initiated to reach the remote destinations. Zone Routing Protocol (ZRP), proposed by Hass *et al.* (2002), applies this idea to create a routing zone around each network member. During the reactive path building phase, if the destination is within the routing zone, a path replay is sent back to the source. Hence, the transmission delay is reduced by comparison with the typical reactive approach. Nevertheless, since the search for the destination is directed through routing zones, the path produced by this protocol is not always the shortest.

Joa and Lu (2002) proposed Zone Based Hierarchical Link State (ZHLS) which is also a hybrid protocol that

creates routing zones based on the geographical locations of the network members. Depending on the positions of the network members, the network is arranged in a form of multiple routing zones. Furthermore, the nodes that belong to the same zone create paths between each other and the links between the routing zones are announced in the entire network. During the reactive search for the destination, the transmission delay is reduced since the route replay is sent once the zone, to which the destination belongs to, is found. On the other hand, the geographical routing zones of this protocol cannot merge. Figure 2 is an example of how ZHLS arranges the nodes in groups based on their positions.

Protocols that organize the network in a hierarchical manner are very similar to the hybrid protocols. The difference between these two strategies is that the hierarchical organization of the network elects a subset of the network responsible for the routing operations whereas in the hybrid protocols, all the network members contribute uniformly.

Pei *et al.* (1999) designed Hierarchical State Routing (HSR) which is a protocol that arranges the network into multiple hierarchical levels. To apply this protocol, each Cluster Head at level N becomes a member in a cluster at level N+1. Afterwards, the network members are identified by the sequence of Cluster Heads IDs relaying them to the root cluster. Consequently, Gateways have multiple IDs since they can be reached by at least two Cluster Heads.

In Cluster Based Routing Protocol (CBRP), designed by Jiang *et al.* (1999), the mobile hosts advertise periodically their neighborhood table. On the basis of this information, a role is attributed to each network member (Cluster Head, Gateway or Cluster Member). Additionally, the reactive method is applied when the destination does not belong to the same cluster. Accordingly, the route request is forwarded between clusters until the destination is located. Moreover, Vidal and José (2010) proposed an extension of this protocol to provide QoS for the new data flow. Cluster Based QoS Routing Protocol (CBQRP) includes the minimum bandwidth required in the route requests. Essentially, the idea in this protocol is to forward the route request by only the Cluster Heads and Gateways that have enough remaining bandwidth to support the new data flow.

Theoretical Analysis

In addition to the advantages/inconveniences that the protocols inherit from their path building strategy, each protocol incorporates other methods that can improve its performance. For instance, the reactive protocols can reduce the transmission delay by adding a local path repair method. Also, the performances of the routing protocols depend on several aspects such as the network members mobility, network size/density, the user application requirements,...etc.

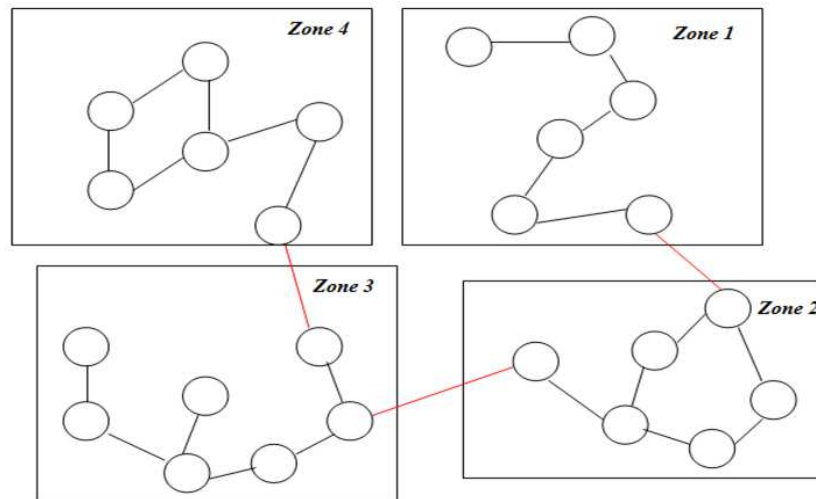


Fig. 2. Zone based hierarchical link state

Table 1. Different protocols characteristics

Parameters Protocol	Resource consumption	Topological awareness	Algorithm complexity	Performance dependency	Advantages
DSDV	Medium	Medium	Medium	Network size/density and the mobility of the nodes.	Fast convergence and routing loop avoidance.
FSR	Medium	Low	Medium	Network density and the mobility of the nodes.	Adaptability to large topologies.
DREAM	High	High	Medium	Network size.	Adaptability to dense topologies. High throughput and retransmission avoidance.
OLSR	High	High	High		
Babel	High	High	Medium	Network size and density.	Low overhead and routing loop avoidance.
BATMAN	High	Medium	Medium		
AODV	Very low	Low	Low	Network size, the number of transmission requests and the mobility of the nodes.	Routing loop avoidance.
DSR	High	Low	Very Low		
AOMDV	Medium	Medium	Low	Network size and the number of transmission requests.	Adaptability to path rupture.
TORA	Medium	Medium	Medium	The number of transmission requests. Network size, the number of transmission requests and the mobility of the nodes.	Transmission delay reduction.
LAR	Low	Medium	Low		
SSR	Low	Low	Low	Network size and the number of transmission requests.	Retransmission avoidance.
ABR	Low	Medium	Low	The number of transmission requests and the network density.	Path rupture avoidance.
ZRP	Medium	Low	Low		Path discovery acceleration.
ZHLS	Medium	Low	Low		
HSR	Medium	Low	Medium		
CBRP	Medium	Medium	Medium		

For example, the proactive protocols are more suitable when the user application needs to maintain the communication between all the network members. In a collaborative system where each user must be aware of the other teammates status, establishing the paths automatically between the components of the network, is crucial. Whereas the proactive protocols automatically create the paths and update them, frequent topological changes force the reactive protocols to restart the path building process from the beginning. On the other hand, the reactive protocols are usually fairly simple to execute

and require very little resources. Table 1 summarizes the characteristics of the MANET routing protocols from a theoretical point of view.

Proposed Algorithm

To achieve proactive routing, the entire network must advertise periodically the collected routing information in a form of control messages. After that, the received control messages are processed by the neighbors to update their topological view. Usually,

the proactive protocols broadcast the entire routing table after a short period of time to avoid routing anomalies and refresh the routing view of the neighbors. Another shared mechanism between the proactive protocols is that all the control messages received from the neighbors must be processed.

In a small/medium size network, the proactive protocols can perform very well without a noticeable effect on the network resources. On the other hand, in a large network, the sizes of the control messages grow as well which directly increases the consumption of the network members resources. Our idea is to reduce the overhead of the proactive protocol by limiting the broadcast of the entire routing table. Instead of frequently advertising the full routing table, the network members broadcast the routing updates. Evidently, in order to announce all the available paths, the broadcast of the global topological view is a must in any proactive protocol. However, the periodical broadcast of the routing table is wasteful in terms of energy. Especially in the network areas where there is no significant changes to be announced.

To enable the protocol to converge adequately while generating an acceptable quantity of overhead, the algorithm that we propose broadcasts the full routing table only if at least one new essential neighbor is sensed. This event driven mechanism significantly reduces the effect of the network size on the quantity of overhead produced by the routing protocol. The protocol that we present in this study is called Effective Routing Based on Overhead Reduction (ERBOR). Initially, we started designing this protocol in our preliminary research on the MANET and the existent Ad hoc routing protocols (Bambrik and Didi, 2013).

Another topological parameter that can influence the proactive protocol is the network density. Constantly processing all the control messages received from the neighbors can consume a large portion of energy and computation time. In Fig. 3 for example, MH1 is located in a dense area of the network and must process all the control messages received from its neighbors. The MPR method included in OLSR improves the protocol adaptability to dense topologies by reducing the number of links announced in network. Still, the network members treat all the control messages received from both the neighbors and MPRs.

Understandably, the amount of energy spent by a network member on routing operations strongly depends on the volume of routing information processed by it. For that reason, we included a modified MPR selection mechanism that enables ERBOR to disregard control messages that contain redundant routing information. Instead of processing all the received control messages like OLSR, ERBOR verifies first the control message generator field to determine if the message is useful. Simply, when the generator of a control message is reachable through an existing path, the control message is ignored. Otherwise, the control message is fully processed.

Control Messages Processing

ERBOR is based on three tables and the MPR list. The first table is the Routing Table (RT) which contains for each path: (a) the destination ID, (b) next hop, (c) hop count. The second table is the Next Update Table (NXUT) in which, the updates that have not been advertised yet, are stored. Each update contains: (a) the destination ID, (b) hop count, (c) type of the update (a deletion or path entry). The third table is called Delayed Update Table (DUT) and it contains all the updates that need to be checked before they can be advertised. In case an entry from the DUT is removed before the next control message broadcast, a new removal update is included in the NXUT to announce that the corresponding route is no longer valid. Otherwise, if an entry in the DUT still exists after the subsequent broadcast (the information is still valid), a positive update is added to the NXUT.

The network members periodically broadcast their NXUT and clear them afterwards. Then, between two broadcasts, the control messages received from the neighbors are examined. The control message content can be either the NXUT (recent topological events only) or the full routing table. This approach is slightly similar to the one applied in DSDV with several changes. DSDV broadcasts the full routing table when a significant change is sensed so that the size of next incremental message is reduced. Also, DSDV periodically broadcasts the full routing table of all the network members after a short period. The idea we propose in ERBOR is to broadcast the full routing table of a network member only when it is necessary. Table 2 represents the format of the control messages generated by ERBOR. Also, the *CM_Sequence_Number* is used to detect packet loss.

All the control messages contain a list of updates that can be either a deletion update (negative update) or a path update (positive update). The NXUT contains both types of update. On the other hand, when the control message contains the full routing table, all the entries are positive updates. The algorithm in Fig. 4 shows how the control messages are handled by ERBOR.

Removal Update

A path contained in the routing table can be deleted only if a negative update is received from the corresponding next hop or the link with the next hop is broken. Also, if a path is deleted, all the corresponding temporary path updates in the DUT and NXUT, are deleted as well. Subsequently, for each deleted path, a removal update is added to the NXUT in order to announce the detected change. Another possible situation is that a mobile host receives a removal update for a destination that does not exist in its routing table. Logically, the deletion update is ignored in this scenario. The schema in Fig. 5 explains how this procedure is implemented.

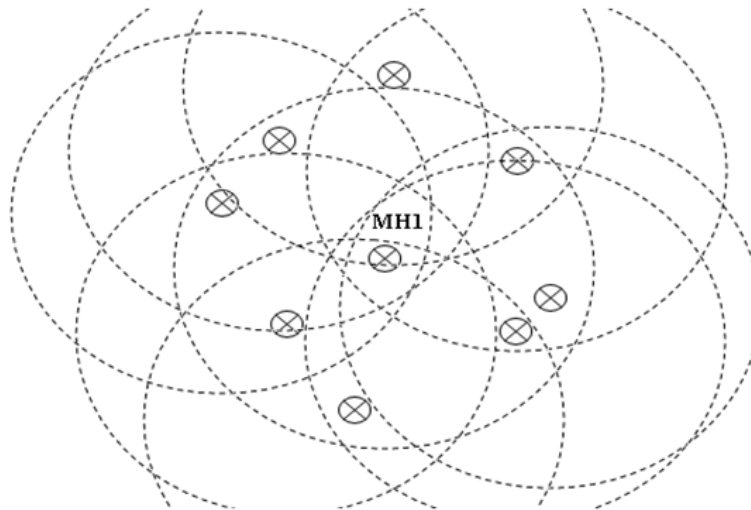


Fig. 3. The effect of the network density

```

public void ProcessingRoutingUpdateMessage (ControlMessage UpdateMessage){
    int i,n=UpdateMessage.Update_List.size();
    int CM_GeneratorID= UpdateMessage.getGeneratorID();
    boolean Skip_Update;
    RoutingEntry DestinationEntry;
    UpdateEntry Update;
    int GeneratorID= UpdateMessage.getGeneratorID();
    if(!C_MSG Is Received From_MPR(UpdateMessage)) return ;
    for(i=n-1;i>=0;i--){
        Update= (UpdateEntry) UpdateMessage.Update_List.elementAt (i);
        Skip_Update=false;
        if(Update.getDestinationID()==ID_Current_node) Skip_Update=true;
        if(isMPR(Update.getDestinationID())){
            Skip_Update=true;
            if(!Update.getUpdateType())InsertInDUT(new UpdateEntry (true,Update.getDestinationID(),1));
            else if (Update.getHopCount()==1)PatchAllPaths (Update.getDestinationID(),CM_GeneratorID);
        }
        if(!Skip_Update){
            DestinationEntry=GetCorrespondingRoutingEntry (Update.getDestinationID());
            if(DestinationEntry!=null){
                if(!Update.getUpdateType()){
                    Process_Delete_Update(Update, DestinationEntry, CM_GeneratorID);
                }else {
                    Processing_Update_For_An_Existing_Path(Update, DestinationEntry, CM_GeneratorID);
                }
            }else Creating_A_New_Routing_Etry(Update, DestinationEntry, CM_GeneratorID);
        }
    }
}
    
```

Fig. 4. Control message processing algorithm

Table 2. Control message format

CM_Generator_ID	CM_Sequence_Number	CM_Type
Destination_ID 1	Hop_count	Update_Type
Destination_ID 2	Hop_count	Update_Type
.	.	.
Destination_ID n	Hop_count	Update_Type

On the other hand, if the next hop is not the originator of the removal update, the path in the routing table can be an alternative path unaffected by the received update. To verify that the available path

is still valid, the path in the routing table is added to the DUT. After the next broadcast, this alternative path is included in the NXUT, if no removal update is received for it. Afterwards, the path is advertised to notify the originator of the removal update that an alternative path is available.

To explain how a deletion update is handled, we consider the scenario in Fig. 6. When the link between MH5 and MH3 is lost, MH5 generates a negative update to inform the neighbors about the topological change. Let's suppose that for MH1, the next hop

toward MH3, is MH2. After the reception of the update generated by MH5, MH1 does not remove the path attributed to MH3 because, MH5 is not the next hop toward it. Instead, MH1 adds the path describing MH3 to its DUT so it can be verified first and advertised afterwards. This way, MH5 will be informed about the existence of an alternative path toward MH3.

Path Update

A path update, for an existing path, is applied by a network member only if it is received from the corresponding next hop or it has a better metric than the one available in the routing table. Additionally, when a path update describing a new destination is received, this new path is accepted only if, there is no path to the same destination, recently deleted. To

verify this condition, the receiver searches for a negative update describing the corresponding destination in the NXUT.

In case a network member receives a path update with a higher hop count by comparison with the one in its routing table, the receiver creates an update describing the available path and adds it to the DUT. The path in the routing table is verified first then added to the NXUT in case it is still valid. The schemas in Fig. 7 and 8 represent the processing methods for the positive updates. In Fig. 9, MH5 detects MH6 as a new neighbor and generates a path update describing this new link. When MH1 processes this information, it first verifies if there is already a path leading to MH6. In this example MH6 is a direct neighbor for MH1 which means that MH1 has a shorter path than the one advertised by MH5.

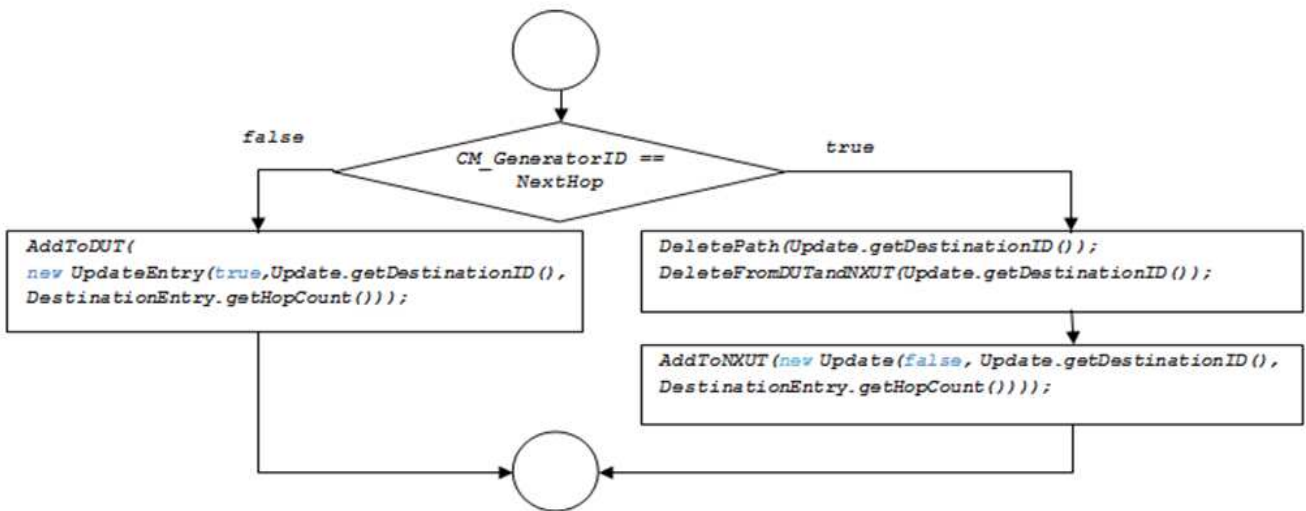


Fig. 5. Deletion update treatment algorithm

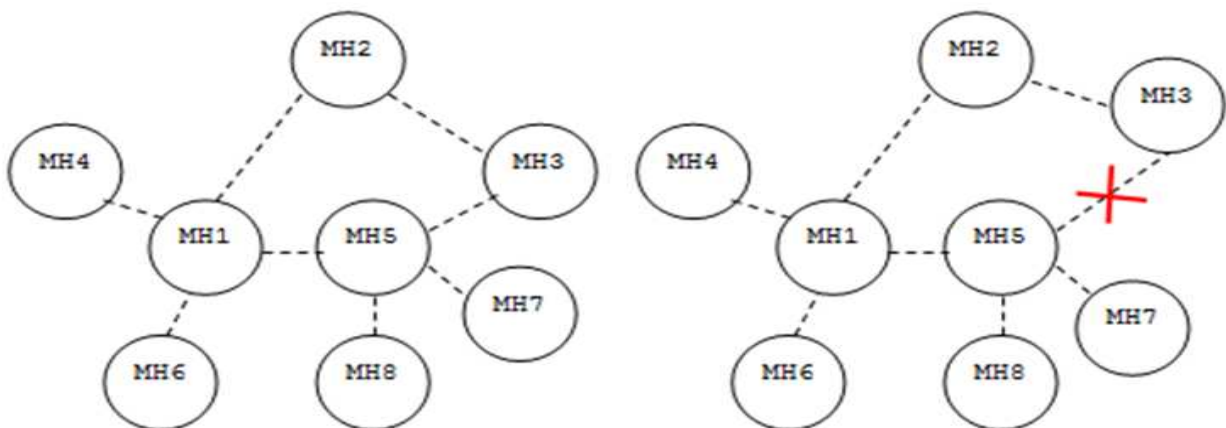


Fig. 6. Deletion update scenario

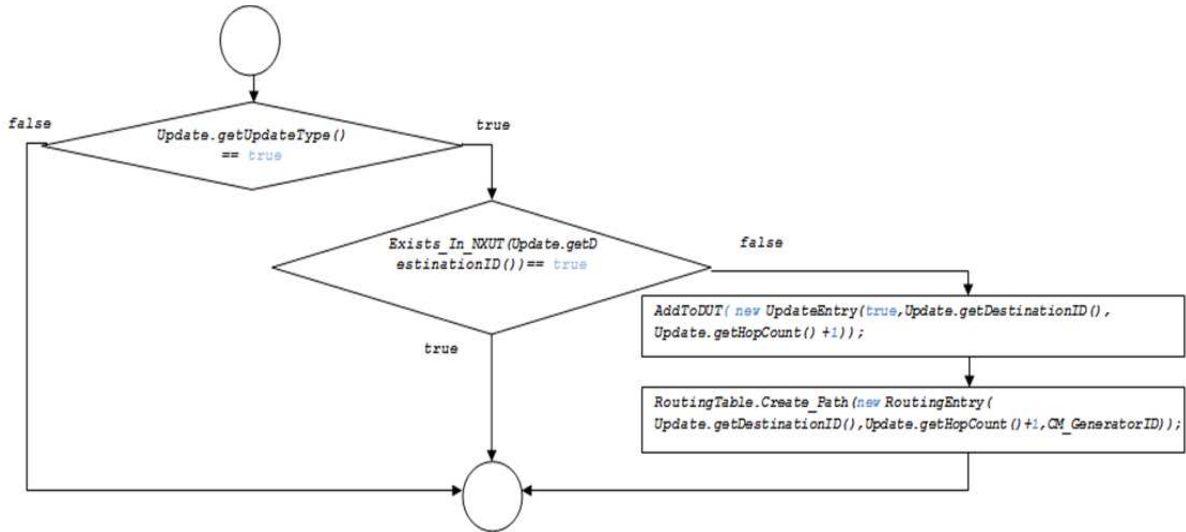


Fig. 7. Creating a new routing entry

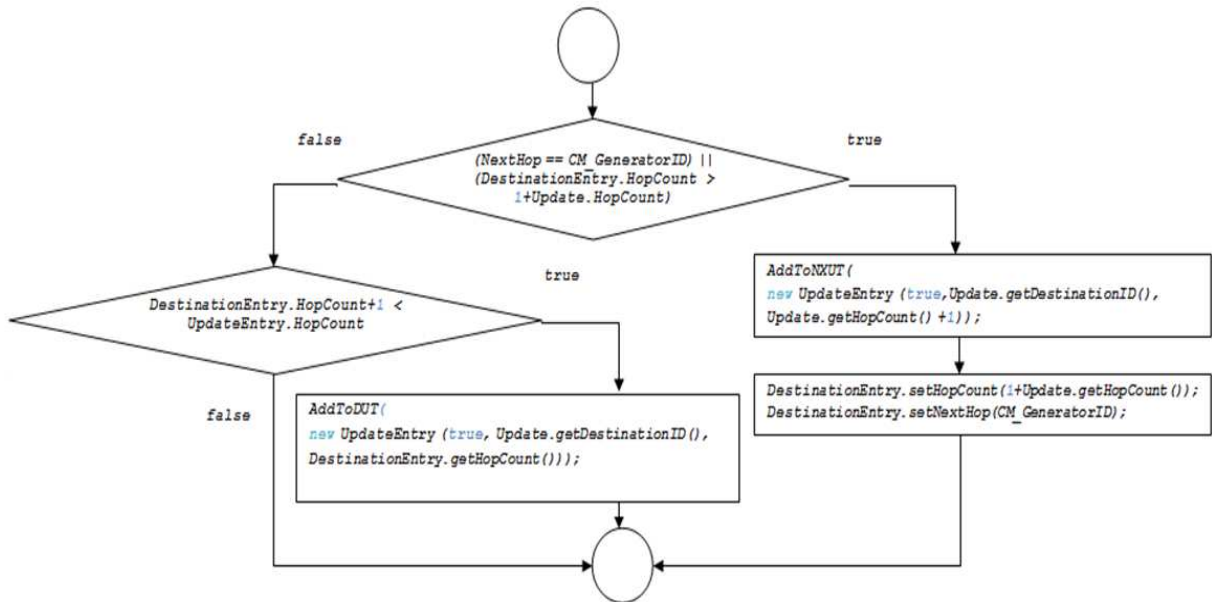


Fig. 8. The algorithm for processing a positive update for an existent path

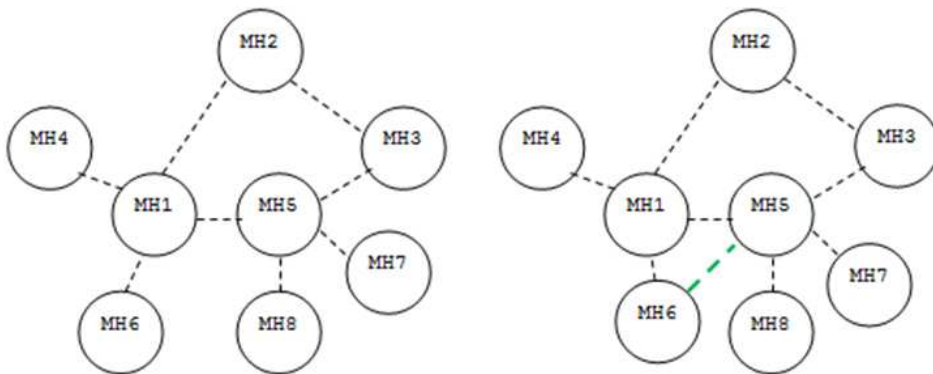


Fig. 9. Path update example

MPR Selection

In a situation where a node has many neighbors, the routing information advertised by some neighbors can be redundant. For example in Fig. 10, MH1 can obtain almost the same routing information from MH8 and MH5. Therefore, it is more efficient to select a subset of neighbors for each network member, that enables the routing protocol to create a complete view of the network. In OLSR, the MPR selection method defines a subset of the neighborhood that enables a mobile host to reach all the neighbors within two hops distance. Nonetheless, the MPRs are used to forward routing information only while the network members process all the received control messages. Our idea is to select a subset of neighbors that enable the protocol to build a path to every potential destination. Afterwards, only the control messages received from the selected neighbors, are treated.

In ERBOR, the receiver of a control message first verifies if the source can be reached through an active MPR. If that is the case, the source of the update is considered as a neighbor only and, the control message received from it, is ignored. Otherwise, if the source cannot be reached by any existing MPR, this neighbor becomes an MPR itself. This way, only the control messages received from the selected MPRs are processed. Accordingly, this method reduces the energy and computation time spent on routing operations. Moreover, only the detection of a new MPR can trigger

the broadcast of the full routing table. Figure 11 shows how the network member detects a potential new MPR and, how the control messages received from the redundant neighbors, are ignored.

When two nodes detect each other as MPRs, they broadcast their entire routing table. The receiver of the routing table compares it with its routing table to create a list of paths that are unavailable for the new MPR. Then, this list is added to the DUT so that it can be verified first then advertised. Afterwards, when a neighbor is selected as an MPR, the full routing tables received from it, are processed only to verify if a path has been removed.

This MPR strategy can function better if the MPR list of each node is minimized. To achieve this, the network members remove any MPR reachable through another MPR. When a new positive update for a current MPR node is received, this new path is added to the RT and the corresponding MPR state is set to "inactive". Then, all the paths based on the "inactive" MPR are patched through the MPR node that offered this new path and the "inactive" MPRs are removed from the MPR list. This method is similar to the one used in hierarchical protocols for Cluster Heads election. For example, in Fig. 12, the movement of MH7 makes it accessible through MH6. Let's suppose that in this example, MH6 is the first to announce this new link. When MH1 detects this change, it patches all the paths based on MH7 through MH6 and removes MH7 from the MPR list. Consequently, MH1 will be processing less routing packets.

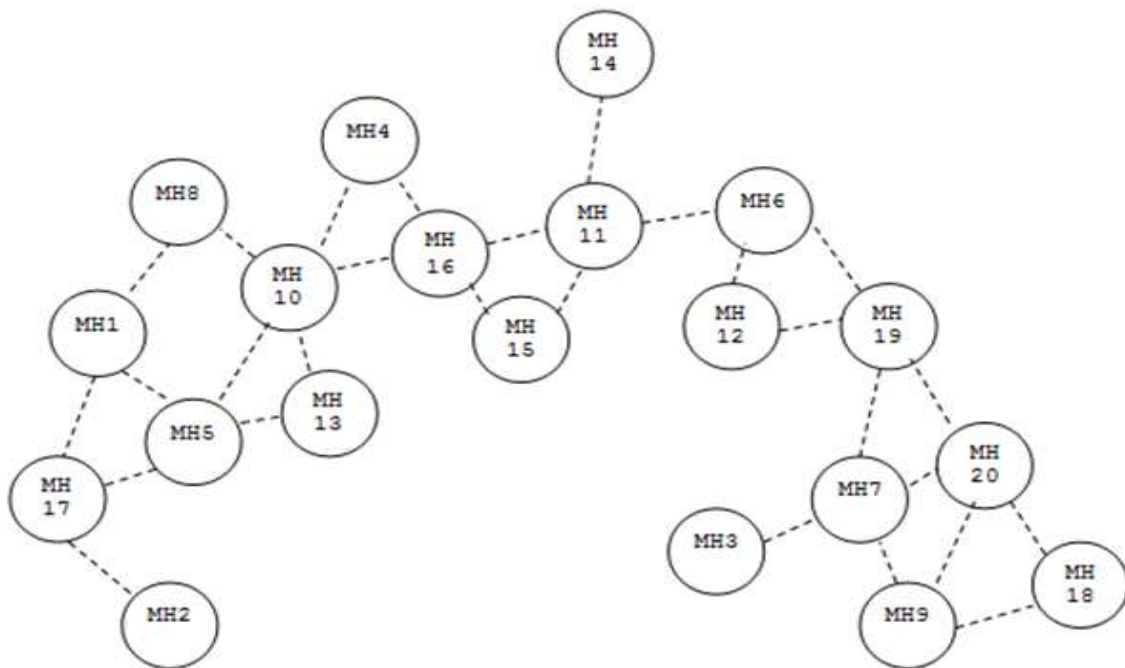


Fig. 10. Redundant routing information treatment

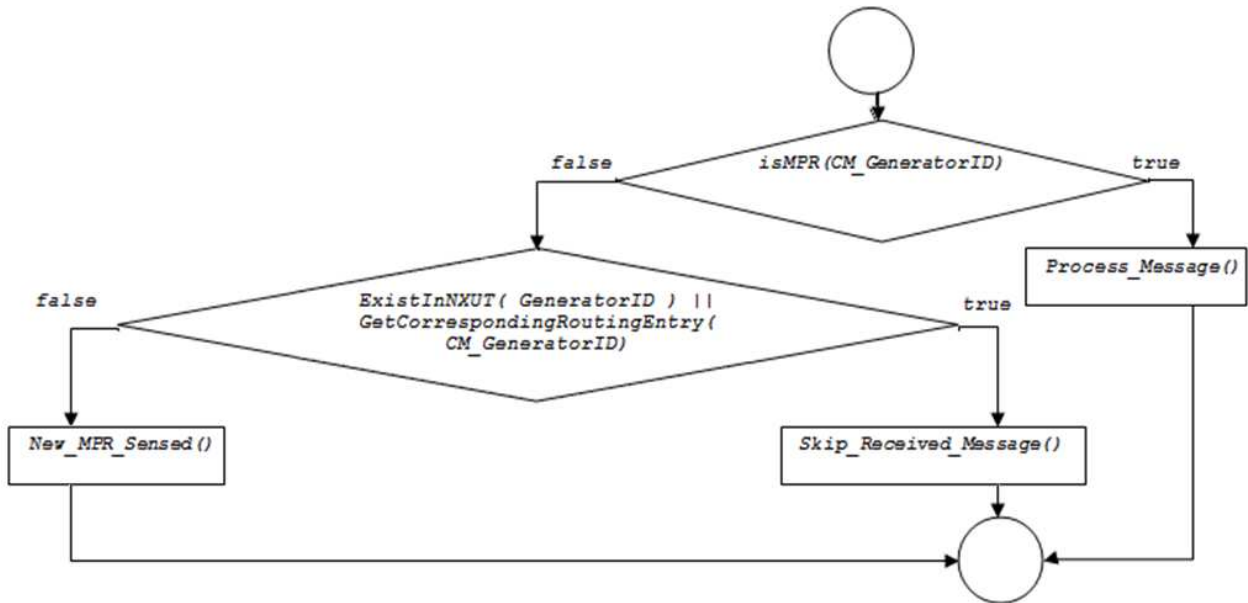


Fig. 11. Control messages sorting algorithm

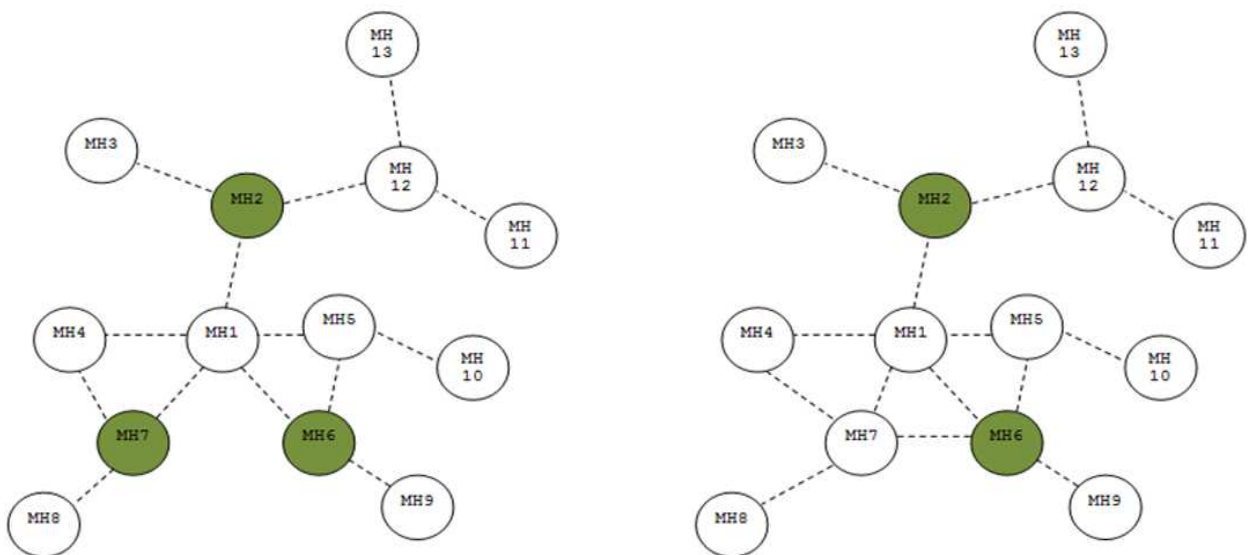


Fig. 12. The MPR set Optimization

Simulation

In order to test ERBOR, we implemented a simulator that can run ERBOR in a dynamic topology. The role of this software is to create a virtual network formed by mobile nodes and execute ERBOR in a distributed manner. First, our purpose was to test the protocol in different topologies. To ensure that the routing protocol works properly in any topology, we created a random movement scenario generator that defines the initial positions of the nodes and the positions to which they are moving toward. The tests were repeated extensively to

ensure that the routing information is advertised correctly and no routing loops can appear.

The routing simulator we implemented, was written in Java (JW, 1995) programming language. Initially, the role of the simulator was to execute the routing functions of ERBOR and display the produced routing tables interactively. Thus, the simulator was meant to operate specifically on the Routing Layer. Afterwards, we added a measurement mechanism that calculates the overhead generated by ERBOR to compare it with the overhead generated if the components of the network advertised their routing tables. Since most of the

proactive protocols advertise the entire routing table after a defined period, the typical proactive method of routing information exchange can be compared with the method that ERBOR applies. To do so, the overhead generated in the network is measured by summing the sizes of the control messages advertised by the network members. Figure 13 displays the different interfaces of the routing simulator.

At the beginning, the tests showed the formation of routing loops when a subset of the network goes out of coverage and renders the network divided into multiple zones. This problem was solved by including the DUT structure instead of the sequence number method. The user is also able to monitor the progression of the routing table in any network member. Both the routing table (RT + MPR list) and the NXUT are displayed proactively. It is also possible to choose a path between the selected node and a destination to display it on the topology interface, as it is shown in Fig. 14. Additionally, the user is able to suspend the node movement or turn off a node in order to test the protocol reaction to node failure. The user can also suspend the movement scenario, restart it or edit the positions of the network members.

In the overhead measurement screen, the simulator indicates the elapsed simulation time, the total overhead generated by ERBOR and the overhead produced by the classical proactive method (the advertisement of the entire routing table). In Fig. 15, the red plot represents the overhead generated by the full routing table broadcast, whereas the overhead generated by ERBOR is displayed in green. At first, the two methods produce

almost the same volume of overhead. This result can be explained by the fact that the network members executing ERBOR advertise their full routing tables when they are starting to establish their MPR list. After the definition of the MPR list, the nodes broadcast more often the routing updates which decrease the overhead. Eventually, when the topology is stable, ERBOR generates only small control messages to maintain the relationship between a node and its selected MPRs. The network in this example was composed of 300 nodes with a transmission range of 250 m for each.

As previously explained, proactive protocols require that all the network members process the entire routing information received from the neighbors. This method costs expensively in terms of energy when the volume of routing information processed is increased. To test the method included in ERBOR, we implemented a frame that displays the volume of routing information processed (as it is displayed in Fig. 16) across the network when: (a) the entire routing table is received and treated (represented by the plot in red), (b) ERBOR treats all the control messages received from the neighbors (represented by the plot in blue), (c) ERBOR treats the control messages received from the MPRs only (represented by the plot in green). The example in Fig. 16 shows that the method of control messages treatment included in ERBOR reduces significantly the volume of routing information treated by the network members. By doing so, this method reduces also the network members CPU usage and the energy spent on routing information processing.

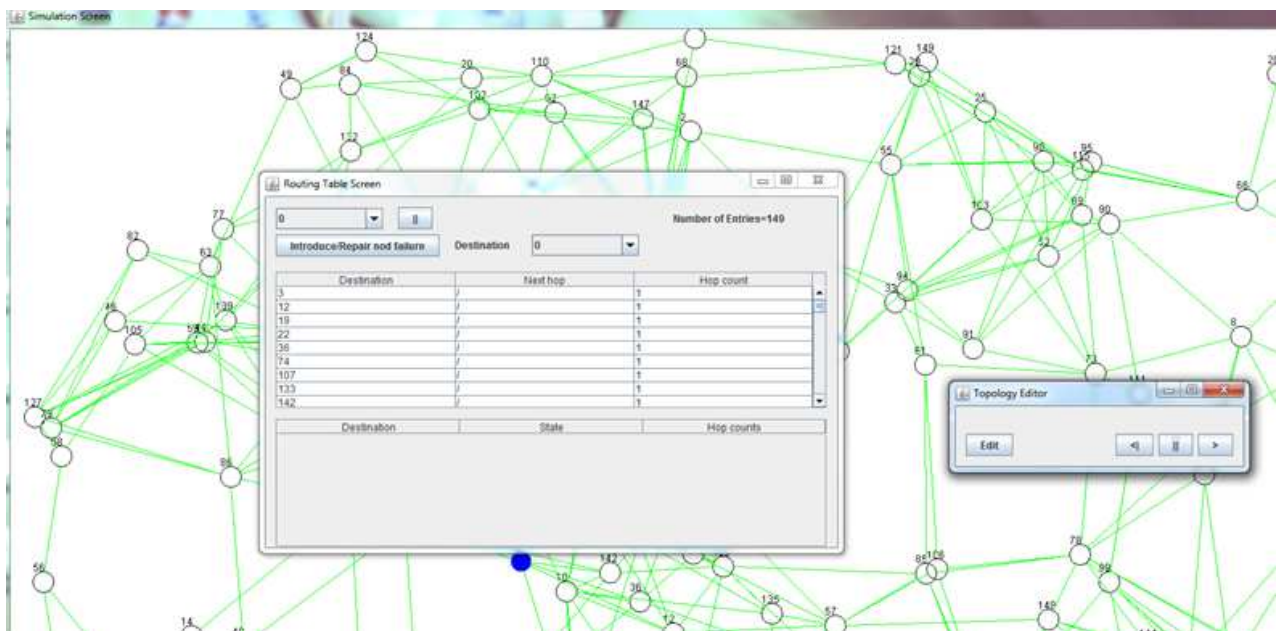


Fig. 13. The routing simulator

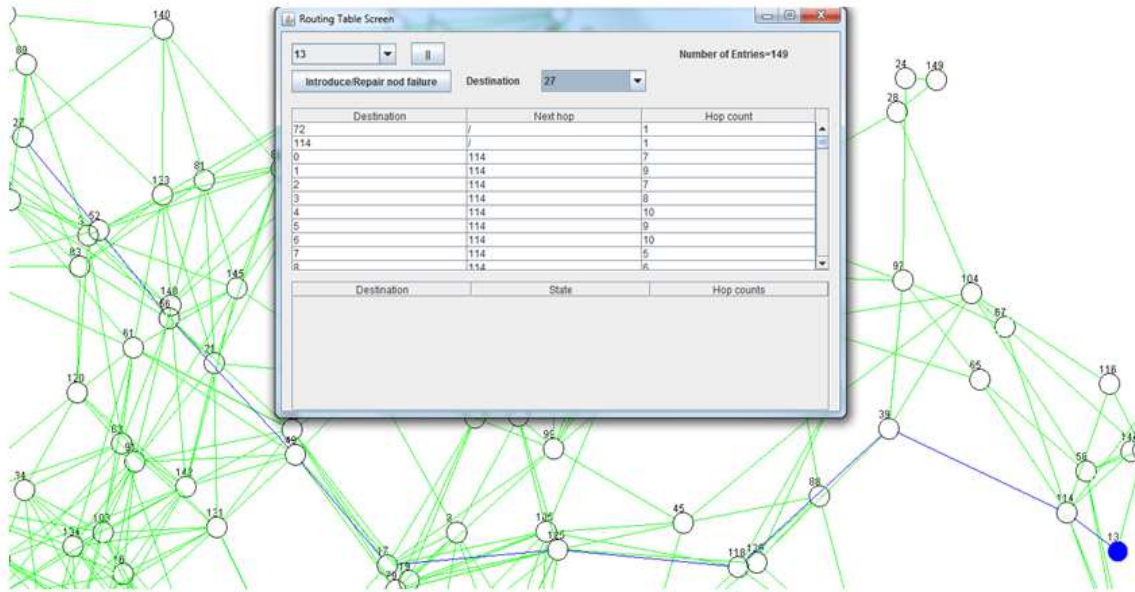


Fig. 14. Path display on the topology screen

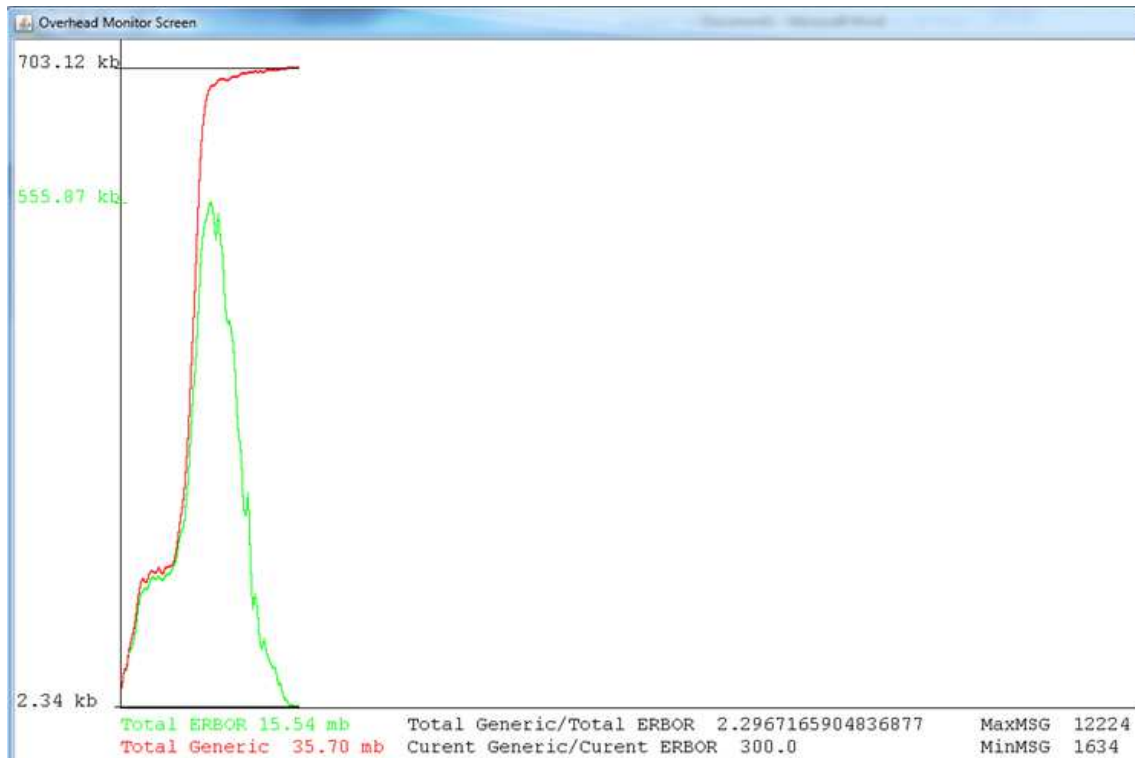


Fig. 15. The overhead measurement screen

In order to facilitate the usage of the simulator, we added an interface that enables the user to define the network topology and a simple movement scenario as it is shown in Fig. 17. The user can also define other parameters like the broadcast period of the control messages and the network members speed of movement. To generate a random topology, the user has to specify

the network size and click on “Generate random topology”. Moreover, the user can observe the evolution of the topology as it is shown in Fig. 18. At first, this frame was used to make sure that all the nodes detect their neighbors. After implementing the MPR selection method, the MPR connection between two nodes was displayed by a green link.

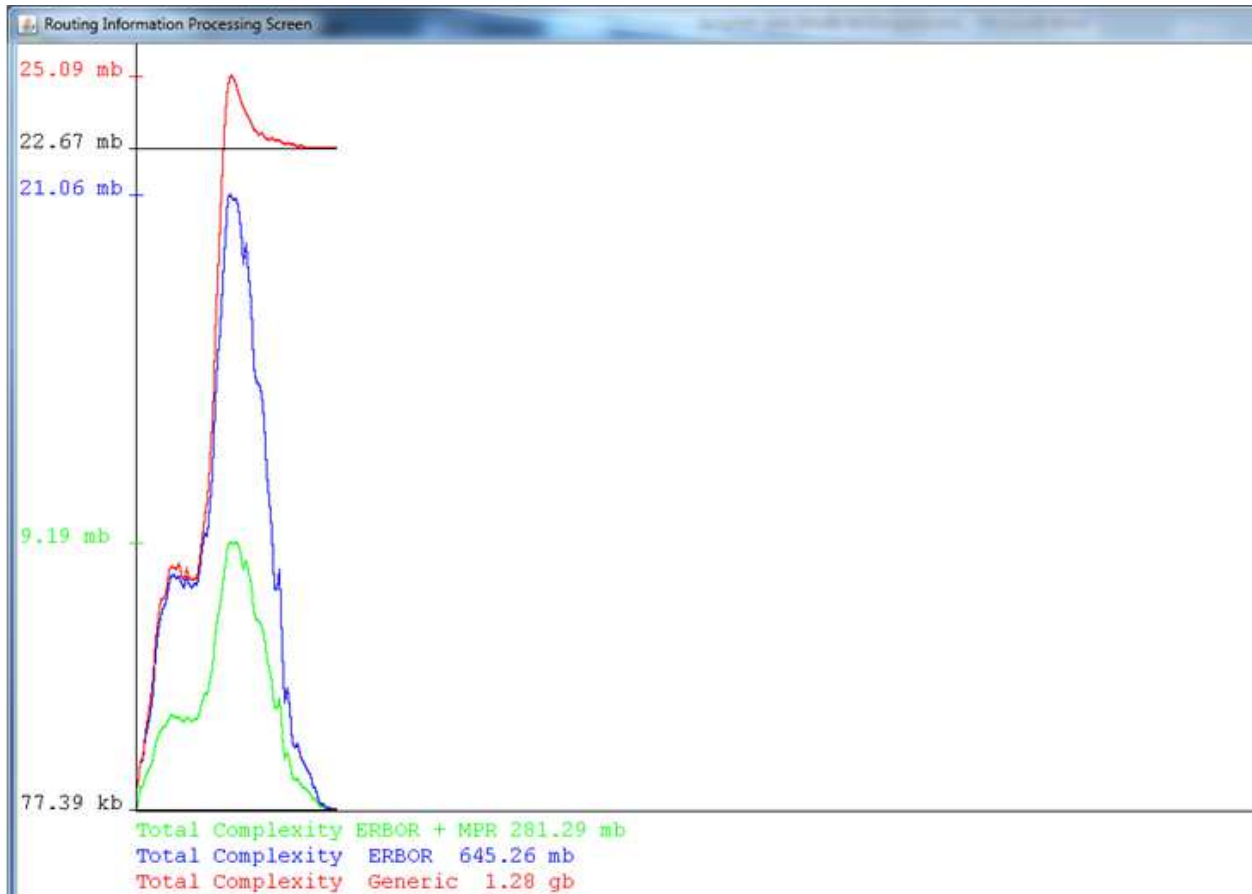


Fig. 16. The volume of routing information processed in the network

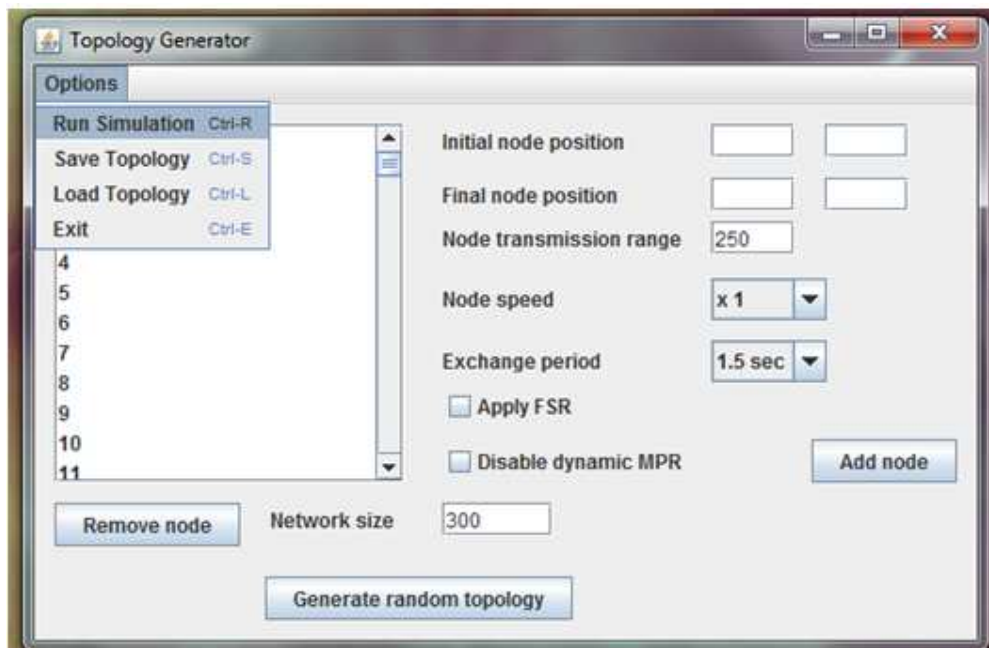


Fig. 17. The topology generator interface

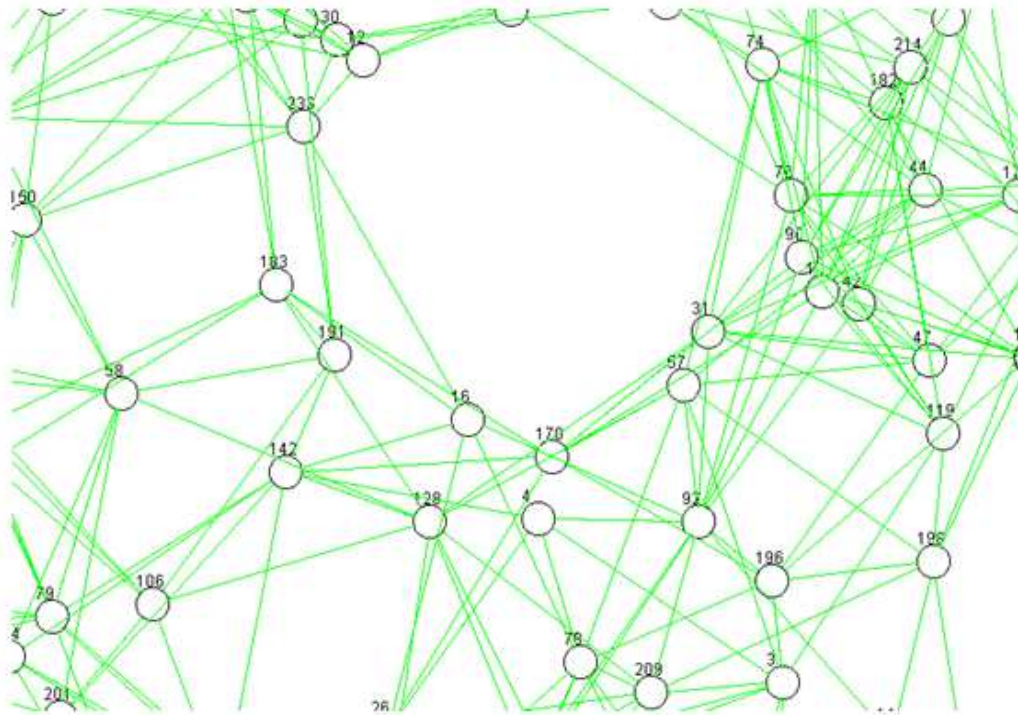


Fig. 18. Topology display

Although this simulation tool is focused only on the routing function, the ability of running the simulation dynamically is a significant advantage. First, the primary role of the simulator is to locate any malfunction of the algorithm which is an easy task when the routing information is displayed interactively. Subsequently, any routing loops/errors can be located. For instance, it is possible to simulate all the network layers with Network Simulator 2 (NS2, 1995), but it is difficult to locate an unusual behavior of the protocol through interpreting a large trace file. Therefore, our focus was to test the protocol in an interactive manner and to make sure that it can operate correctly.

Results

The main disadvantage of all the proactive protocols is their inadaptability to large/high density topologies. Previous studies have illustrated that the increase of the network size, leads directly to the exponential growth of the proactive protocols overhead. For instance, Johann *et al.* (2004) explain that the large overhead produced by the proactive protocols is caused by the frequent broadcast of the entire routing table. Consequently, the resulting overhead is correlated with the value $Network_Size^2$. Furthermore, the obtained results by Deepak *et al.* (2013) and Kavita and Abhishek (2011) indicate that the increase of the network size affects negatively the reactive/hybrid protocols less than the proactive protocols.

Besides, from a network density prospect, the treatment of a large quantity of routing information can exhaust the network resources at high rate. This aspect affects significantly the MANET members since they are usually equipped with a limited computational capacity. To compare our solution, in this context, with other protocols from the same category, we conducted several series of simulations with DSDV, FSR (Sven, 2005) and ERBOR. The simulation parameters were changed in order to measure the overhead when: (a) the network size is expanded, (b) the topology density is raised, (c) the movement speed of the nodes is raised. The results recorded during the simulations are presented in Fig. 19-24. Generally, ERBOR generates the lowest overhead during the tests and treats less routing information than FSR and DSDV.

Discussion

The Network Growth Influence

To test the influence of the network growth on the proactive protocols, we ran a series of simulations where the network size is augmented gradually. The obtained results in Fig. 19 and 20 show how the overhead and the routing information processed by the tested protocols, are influenced. As expected, FSR generates less overhead and treats less routing information than DSDV.

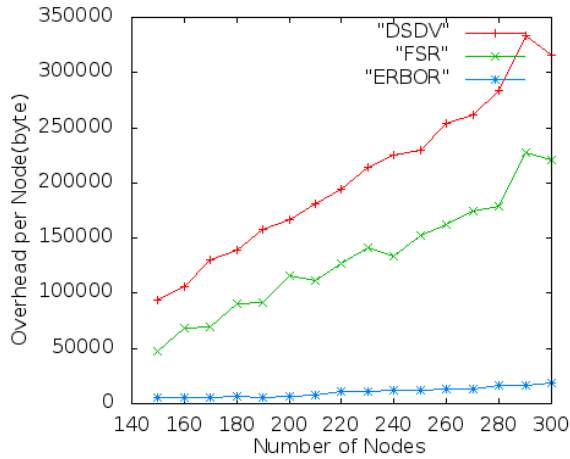


Fig. 19. The impact of the network growth on the overhead generated per node

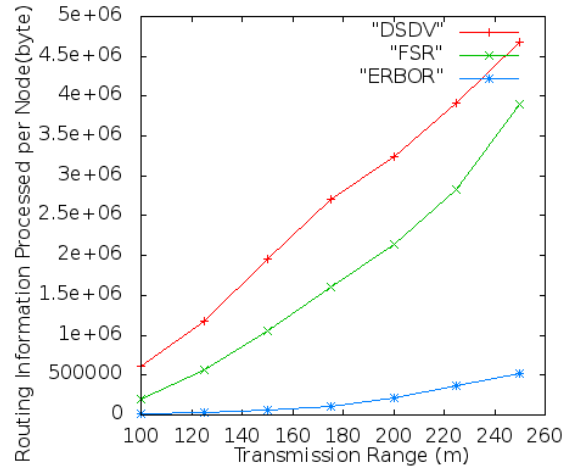


Fig. 22. The impact of the transmission range on the quantity of routing information processed per node

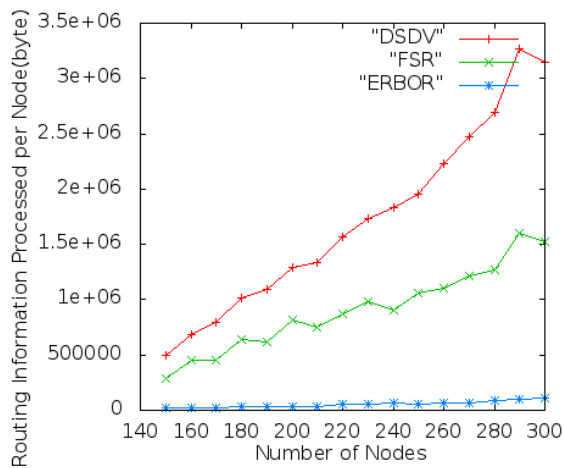


Fig. 20. The impact of the network growth on the quantity of routing information processed per node

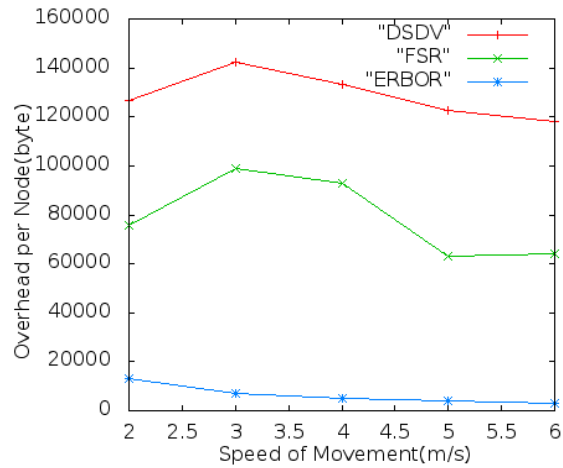


Fig. 23. The impact of the speed of movement on the overhead generated per node

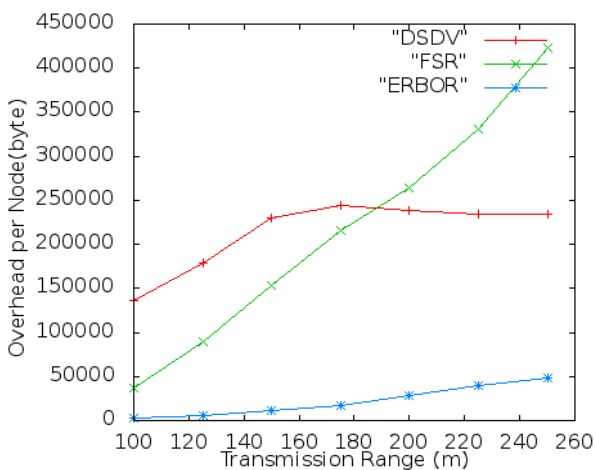


Fig. 21. The impact of the transmission range on the overhead generated per node

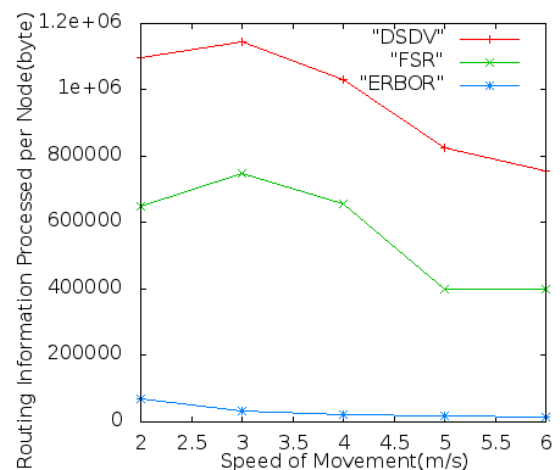


Fig. 24. The impact of the speed of movement on the quantity of routing information processed per node

Although the Fisheye algorithm improves considerably the adaptability of the protocol to large topologies, increasing the network size leads eventually to the increase of the network density. As a result, the size of the neighborhood description, advertised by FSR, is increased as well. The parameters of this experiment are declared in Table 3.

Besides the periodic broadcast of the global topological information, DSDV is based on an event driven method to announce the important topological changes. Although this approach has its benefits from a convergence perspective, this method elevates the overhead when the network members are still in motion. Moreover, the systematic advertisement of the entire routing table increases the overhead as well. Hence, the massive volume of routing information processed and the large quantity of overhead when DSDV is applied.

In contrast to the two previous protocols, the growth of the network appears to have very little influence on ERBOR. From an overhead point of view, the results can be explained by the fact that ERBOR limits the quantity of overhead by advertising the entire routing table of a node, only when a useful connection is established. Thus, the increase of the network size does not directly augment the sizes of the control messages. Moreover, when the topology is stable, FSR and DSDV continue to broadcast the routing messages periodically. On the other hand, ERBOR advertises only the detected changes. Accordingly, whereas the overhead/routing information processed of DSDV and FSR continue to expand, ERBOR consumes a negligible quantity of resources when the network members are steady.

As explained earlier, a shared mechanism between the proactive protocols is the periodic treatment of all the received routing messages. Considering the sizes of the control messages (due to the growth of the network) advertised by DSDV and FSR, both protocols process a massive quantity of routing information in order to maintain the topological view. On the other hand, ERBOR treats only the messages received from the selected MPRs. Combined with the way ERBOR restrains the control message size, the method of MPR selection reduces considerably the influence of the network growth on the volume of routing information processed by the network members.

The Network Density Influence

Another parameter that can heavily influence the proactive method, is the network density. To test how this factor can impact the proactive protocols, we ran a series of simulations where the network density is increased by augmenting the transmission range of the network members. Consequently, the number of neighbors sensed by a node, is augmented as well. The results in Fig. 21 and 22 illustrate that the increase of the network density leads to the increase of the

overhead and the volume of routing information processed, by all three protocols. Table 4 represents the parameters of this simulation.

As mentioned in the theoretical description of FSR, the strategy of this protocol is to broadcast the description of the adjacent nodes more often than the description of the remote destinations. Thus, when the number of close nodes increases due to the network density, the overhead of FSR expands past the quantity generated by DSDV. On the other hand, the overhead of DSDV is relatively stable because the dissemination method used in this protocol does not depend on the number of the adjacent nodes. More importantly, since all the received routing messages must be processed, the increase of the network density causes the elevation of the volume of routing information processed by both protocols.

Due to the MPR selection method, ERBOR advertises the entire routing table only when it is necessary. Although this method reduces the volume of overhead, the increase of the network density leads inevitably to the increase of the detected MPRs. Nonetheless, the effect of the network density is significantly reduced by comparison with FSR and DSDV. Moreover, unlike FSR and DSDV, only the essential routing messages are processed. Thus, the influence of the network density on ERBOR, is limited.

Another perceptible outcome is that, although FSR generates more overhead than DSDV, DSDV processes more routing information. While this result can seem odd, it can be simply traced to the event driven nature of DSDV. In FSR, the broadcast of the neighborhood description is done periodically. Thus, the increase of the routing messages sizes, increases directly the volume of routing information processed. On the other hand, the broadcast of the routing packets is event driven when DSDV is applied. To lower the overhead, when an important event is detected, DSDV broadcasts the entire routing table to limit the sizes of the subsequent incremental messages. Moreover, due to the increase of the network density, any sensed change deemed as essential, will trigger the immediate broadcast of the routing message which is received by all the adjacent nodes. Hence, the elevation of the routing information processed by DSDV. In Fig. 25, the recorded quantity of routing messages processed by DSDV and FSR, during this series of simulations, are displayed.

Network Members Movement

To test the adaptability of the three protocols based on the nodes speed of movement, we made a series of simulations where the nodes speed of movement is augmented gradually. The obtained results in Fig. 23 and 24, show that DSDV and FSR are influenced randomly. On the other hand, the overhead/routing information processed of ERBOR, decreases. The parameters of this experiment are shown in Table 5.

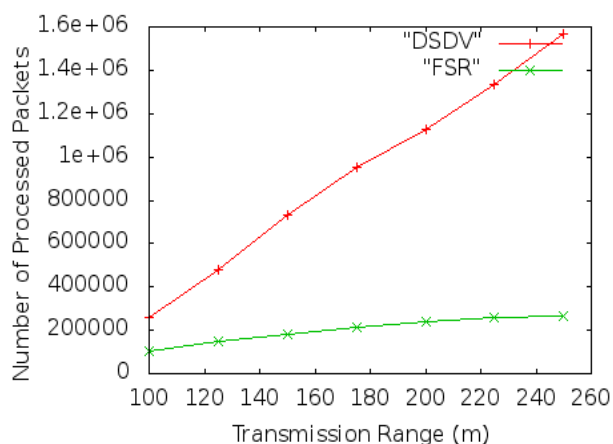


Fig. 25. The overall quantity of routing packets processed by DSDV and FSR

Table 3. The network growth parameters

Parameters	DSDV	FSR	ERBOR
Broadcast period	15 sec	5 sec	5 sec
Simulation space		1500×1500 m	
Transmission range		150 m	
Speed of movement		4 m s ⁻¹	
Network size		150-300	
Simulation time		400 s	

Table 4. The network density parameters

Parameters	DSDV	FSR	ERBOR
Broadcast period	15 sec	5 sec	5 sec
Simulation space		1500×1500 m	
Transmission range		150-250 m	
Speed of movement		4 m s ⁻¹	
Network size		250	
Simulation time		400 s	

Table 5. The network members movement speed parameters

Parameters	DSDV	FSR	ERBOR
Broadcast period	15 sec	5 sec	5 sec
Simulation space		1500×1500 m	
Transmission range		150 m	
Speed of movement		2-6 m s ⁻¹	
Network size		250	
Simulation time		400 s	

Table 6. The topological parameters influence on DSDV, FSR and ERBOR

Parameter	Protocol	Network size	Network density	Node speed of movement
DSDV	Overhead	High	Medium	High
	Processed information	High	High	High
FSR	Overhead	Medium	High	Low
	Processed information	Medium	High	Low
ERBOR	Overhead	Low	Medium	Low
	Processed information	Low	Medium	Low

First, it is obvious that increasing the network members speed of movement does not influence FSR directly. Unlike DSDV, the routing messages advertisement method of FSR is independent from the topological events. As it is shown in Fig. 24, the highest values of overhead are generated by DSDV. Regardless of the difference between the overhead produced by DSDV and FSR, this protocol generates a higher quantity of routing messages when the topology changes frequently.

Unlike DSDV, ERBOR periodically announces only the changes based on the updates received from the MPRs. Thus, the effect of the topological changes on the overhead of ERBOR, is reduced. Obviously, when the network members are moving fast, they reach their final position faster and the topology becomes stable. Since ERBOR announces the routing updates periodically only, the overhead of this protocol is lowered after its convergence. Otherwise, when the network members are moving slowly, the topological events are more continuous. Hence, it is also noticeable that ERBOR generates more overhead when the network members are moving slowly. Table 6 summarizes the results obtained in this section.

Conclusion

After implementing ERBOR and putting it to the test, we obtained positive results that can be used to improve it in different aspects. For instance, a local path repair technique like the one used in ABR can limit the routing update dissemination. Instead of immediately announcing that a link with an MPR is lost, the node announces this routing update in its neighborhood to search for an alternative path. If the path repair process fails, a path removal update is announced. This method can be effective when the network members are moving at slow/medium speed. Otherwise, the route repair attempt will only slowdown the convergence. A simpler idea is to include the FSR method in order to accelerate the convergence without elevating the overhead. In future studies we intend to include other methods to enhance the protocol adaptability to the user application requirements.

Acknowledgement

The authors would like to thank the reviewer for his effort and valuable comments.

Funding Information

The authors of this paper have no financial funding to report.

Author's Contributions

Both authors have contributed equally during this work.

Bambrik Ilyas: Was in charge of: (a) implementing the protocol and the development of the simulator, (b) interpreting the results.

Didi Fedoua: Contributed in: (a) the study of the related works, (b) protocol testing.

Ethics

The corresponding author states that this paper is original. Both authors have read the paper and approved of its content.

References

- Alshanyour, M. and U. Baroudi, 2008. Bypass AODV: Improving Performance of Ad Hoc On-Demand Distance Vector (AODV) Routing protocol in wireless ad hoc networks. Proceedings of the 1st International Conference on Ambient Media and Systems, Feb. 11-14, Quebec, Canada.
DOI: 10.4108/ICST.AMBISYS2008.2895
- Andreas, T., L. Thomas, G. Hannes, P. Bernd and K. Aaron *et al.*, 2009. OLSR development.
- Bambrik, I. and F. Didi, 2013. Effective routing based on overhead reduction. *Int. J. Applic. Innovat. Eng. Manage.*, 2: 102-110.
- Basagni, S., I. Chlamtac, V.R. Syrotiuk and B.A. Woodward, 1998. A distance routing effect algorithm for mobility (DREAM). Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Oct. 25-30, Dallas, TX, USA, pp: 76-84.
DOI: 10.1145/288235.288254
- Chandramohan, K. and P. Kamalakkannan, 2014. Traffic Controlled-Dedicated Short Range Communication: A Secure Communication Using Traffic Controlled Dedicated Short Range Communication Model In Vehicular Ad Hoc Networks For Safety Related Services. *J. Comput. Sci.*, 10: 1315-1323.
DOI: 10.3844/jcssp.2014.1315.1323.
- Clausen, T. and P. Jacquet, 2003. Optimized link state routing protocol. IETF Internet Draft.
- Day, K., A. Touzene, B. Arafah and N. Alzeidi, 2011. Parallel routing in mobile ad hoc networks. *Int. J. Comput. Netw. Commun.*, 3: 77-94.
DOI: 10.5121/ijcnc.2011.3506
- Deepak, K.P., K. Rakesh and A.K. Daniel, 2013. Performance analysis and behavioral study of proactive and reactive routing protocols in MANET. *Int. J. Adv. Res. Comput. Commun. Eng.*, 2: 1789-1796.
- Dorigo, M. and L.M. Gambardella, 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evolutionary Computat.*, 1: 53-66.
DOI: 10.1109/49.779923
- De Couto, D.S.J., A. Daniel, J. Bicket and M. Robert, 2003. A high-throughput path metric for multi-hop wireless routing. Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, Sept. 14-19, San Diego, CA, USA, pp: 134-146. DOI: 10.1145/938985.939000
- Dube, R., C.D. Rais, K.Y. Wang and S.K. Tripathi, 1997. Signal stability-based adaptive routing for ad hoc mobile networks. *IEEE Personal Commun. Magazine*, 4: 36-45. DOI: 10.1109/98.575990
- Gerla, M., G. Pei, H. Xiaoyan and T.W. Chen, 2002. Fisheye state routing protocol for ad hoc networks. IETF Internet Draft.
- Gopalakrishnan, S. and P. Ganeshkumar, 2015. Secure and Efficient Transmission in Mobile Ad hoc Network to Identify the Fake ID's Using Fake ID Detection Protocol. *J. Comput. Sci.*, 11: 391-399.
DOI: 10.3844/jcssp.2015.391.399
- Hass, Z.J., R. Pearlman and P. Samar, 2002. The Zone Routing Protocol (ZRP) for ad hoc networks. IETF Internet Draft.
- JW, 1995. Sun Microsystems.
- Jiang, M., J. Ji and Y.C. Tay, 1999. Cluster based routing protocol. National University of Singapore.
- Joa, N.M. and I.T. Lu, 2002. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. *IEEE J. Selected Areas Commun.*, 17: 1415-1425. DOI: 10.1109/49.779923
- Johann, L., M.B. José and G.V. Jorge, 2004. Analysing the overhead in mobile ad hoc network with hierarchical routing structure. Technical University of Catalonia.
- Johnson, D., Y. Hu and D. Maltz, 2007. The Dynamic Source Routing Protocol (DSR) for mobile ad hoc networks for IPv4. RFC.
- Juliusz, C., 2010. The babel routing protocol. IETF Internet Draft.
- Kanniche, H., F. Louati, M. Frikha and F. Kamoun, 2011. A QoS routing protocol based on available bandwidth estimation for wireless ad hoc networks. *Int. J. Comput. Netw. Commun.*, 3: 219-239.
DOI: 10.5121/ijcnc.2011.3114
- Kavita, P. and S. Abhishek, 2011. A Comprehensive Performance Analysis of Proactive, Reactive and Hybrid MANETs Routing Protocols. *Int. J. Comput. Sci.*, 8: 432-441.
- Ko, Y.B. and N.H. Vaidya, 2000. Location aided routing in mobile Ad hoc networks. *Wireless Networks J.*, 6: 307-321. DOI: 10.1023/A:1019106118419
- Marina, M.K. and R.S. Das, 2006. Ad hoc on-demand multipath distance vector routing. *Wireless Commun. Mob. Comput.*, 6: 969-988. DOI: 10.1002/wcm.432

- NS2, 1995. Network simulator 2 website.
- Neumann, A., C. Aichel and M. Lindner, 2008. Better approach to mobile ad-hoc Networking. IETF Internet Draft.
- Park, V.D. and M.S. Corson, 1997. A highly adaptive distributed routing algorithm for mobile wireless networks. Proceedings of the 6th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97), Kobe, Japan. pp: 1405-1413.
DOI: 10.1109/INFCOM.1997.631180
- Pei, G., M. Gerla and H. Xiaoyan, 1999. A wireless hierarchical routing protocol with group mobility. IEEE Wireless Communications and Networking Conference (WCNC1999), New Orleans, L.A., USA. 1538-1542.
DOI: 10.1109/WCNC.1999.796996
- Perkins, C.E. and P. Bhagwat, 1994. Highly dynamic destination-sequenced distance-vector routing for mobile computers. ACM SIGCOMM' 94 Conference on Communications Architectures, Protocols and Applications, New York, USA. 234-244. DOI: 10.1145/190314.190336.
- Perkins, C.E., R.E. Belding and S. Das, 2003. Ad hoc On-Demand Distance Vector (AODV) Routing. DOI: 10.17487/RFC3561
- Poonkuzhali, R., S.M. Yakub and S. Annaji, 2014. A preemitive link state spanning tree source routing protocol for mobile ad hoc networks. J. Comput. Sci., 10: 85-90. DOI: 10.3844/jcssp.2014.85.90
- Sensarma, D. and K. Majumder, 2013. An efficient ant based QoS aware intelligent temporally ordered routing algorithm for MANETs. Int. J. Comput. Netw. Commun., 5: 189-203.
DOI: 10.5121/ijcnc.2013.5415
- Surjeet., A. Parkash and R. Tripathi, 2014. Bandwidth constrained priority-based routing algorithm for mobile ad hoc networks. Int. J. Commun. Netw. Syst. Sci., 7: 141-150.
DOI: 10.4236/ijcns.2014.75016
- Sven, J., 2005. FSR Patch for NS2, Sourceforge.NET.
- Toh, C.K., 1997. Associativity-based routing for ad hoc mobile networks. Wireless Personal Commun., 4: 103-139. DOI: 10.1023/A:1008812928561
- Vidal, J.G. and B.M. José, 2010. Cluster based QoS routing protocol for MANET. Int. J. Comput. Theory Eng., 2: 1793-8201.