Paper

A Scalable Multicast Routing Protocol for Mobile Ad-Hoc Networks

Liana Khamis Qabajeh

Faculty of Information Technology and Computer Engineering, Palestine Polytechnic University, Hebron, Palestine

https://doi.org/10.26636/jtit.2022.155121

Abstract—The multicasting technique supports a variety of applications that require data to be instantaneously transmitted to a set of destination nodes. In environments with continuously moving nodes, such as mobile ad-hoc networks, the search for efficient routes from sources to the projected destinations is a common issue. Proposed Windmill protocol provides a scalable multicast solution for mobile ad-hoc networks. Windmill aims to improve routing protocol's performance by introducing a hierarchal distributed routing algorithm and dividing the area into zones. Additionally, it attempts to demonstrate better scalability, performance and robustness when faced with frequent topology changes, by utilizing restricted directional flooding. A detailed and extensive simulated performance evaluation has been conducted to assess Windmill and compare it with multicast ad-hoc on-demand distance vector (MAODV) and on-demand multicast routing protocols (ODMRP). Simulation results show that the three protocols achieved high packet delivery rates in most scenarios. Results also show that Windmill is capable of achieving scalability by maintaining the minimum packet routing load, even upon increasing the nodes' speed, the number of sources, the number of group members and the size of the simulated network. The results also indicate that it offers superior performance and is well suited for ad-hoc wireless networks with mobile hosts. The trade-off of using Windmill consists in slightly longer paths – a characteristic that makes it a good choice for applications that require simultaneous data transmission to a large set of nodes.

Keywords—ad-hoc networks, MAODV and ODMRP, position-based multicast routing protocol, simulated performance evaluation.

1. Introduction

A wireless ad-hoc network is a multi-hop self-organizing structure requiring rapid deployment and dynamic reconfiguration [1], [2]. Each participating node has a wireless interface and communicates with other nodes [3]. One of the most important concerns in ad-hoc networks is related to the routing relied upon to forward data packets to the destination [3], [4]. For example, such a network may be implemented to forward packets to students in a university building, soldiers on a battlefield, participants of a conference, and vehicles on the road [3], [5]. The limited number of power nodes and limited bandwidth of the wireless medium require the power consumption and transmission overhead be reduced [1], [2], [4]. Moreover, efficient routing is of key significance, since all nodes act both as hosts and routers and are usually moving rapidly in most cases [6]–[8].

Multicasting is an ideal communication scheme that efficiently supports a wide variety of applications that require collaboration between nodes [9], [10]. Hence, it supports applications that involve simultaneous data transmission to hosts. Military battlefields, disaster recovery, rescue sites and emergency searched are examples of multicast applications for mobile ad-hoc networks [11]. Multicast group members may move, therefore causing random and rapid topology changes at unpredictable times [12]. Thus, tree reconfiguration schemes and membership information logging techniques should be as simple as possible to ensure reduced channel overhead [7], [8], [13]. The constrained power, limited bandwidth, and mobile hosts make the design of a multicast protocol a challenge [14]. Additionally, the need to rely on scalable energy-efficient protocols, along with the existence of inexpensive and low-power positioning instruments, justify the application of position-based routing in mobile ad-hoc networks [7].

In this paper, the Windmill multicast routing model is presented. It introduces a hierarchical distributed routing algorithm to improve performance of the routing protocol and to distribute load by dividing the area into zones. Additionally, the protocol attempts to offer higher scalability, performance and robustness when faced with frequent topology changes, by relying on the idea of restricted directional flooding. Hence, each group member should keep zone leaders (ZL) of its zone updated about its position.

Windmill consists primarily of five phases: network setup, network maintenance and membership update, route instantiation, route maintenance, and, finally, data transmission. The network setup phase includes dividing the area into zones, deciding on initial ZLs, and assigning the interested nodes to different multicast groups. The network maintenance and membership update phase deals with keeping track of the network's structure during node movements and changes.

Whenever a source node has data to be sent to a multicast group, the route instantiation phase is initiated by sending route request packets, mainly with the use of restricted directional flooding. After finishing route discovery and setup, the source begins the data transmission phase by sending the data to the intended destinations. When needed, the route maintenance phase is conducted to repair the broken routes.

We evaluated the performance of the proposed protocol via simulation and compared it with MAODV and ODMRP. Simulation results show that Windmill offers superior performance, regardless of the nodes' mobility speed, the number of sources, the number of group members and network size. Furthermore, Windmill achieved good scalability by maintaining the minimum packet routing load in all presented scenarios, compared to MAODV and ODMRP. The disadvantage of Windmill has the form of slightly longer paths passing through ZLs. Thus, it is suitable for achieving scalability and reducing the overhead of multicast routing in ad-hoc networks established between students of a university, soldiers on a battlefield, rescuers in a disaster area, and sensor-based IoT networks.

The remaining sections of this paper are organized as follows. Related work in presented in Section 2. Section 3 presents the concept behind the Windmill protocol. Section 4 contains a simulated comparison of Windmill, MAODV and ODMRP protocols. Section 5 discusses our findings. The paper is concluded in Section 6, where future directions are discussed as well.

2. Related Work

Multicast routing protocols are classified based on their delivery structure and ability to maintain connectivity between multicast group members. In [9], the authors classified these routing protocols into six categories: flooding, tree-based, mesh-based, hybrid, hierarchical/adaptive multicast, and location-based. Flooding is the easiest way, since it eliminates the need to maintain explicit infrastructure for multicast forwarding. A source initiates a multicast session by broadcasting the packet to its neighbors. Receiving nodes rebroadcast the packet to their neighbors upon receiving the first copy. This process continues until flooding the packet to the whole network. Hence, such a technique offers the lowest control overheads, it is considered to be the most reliable scheme, and data packets are quickly propagated within the network. However, this comes at the expense of generating considerable data traffic in the wireless environment and wasting bandwidth, especially in large networks [15], [16].

In tree-based protocols, the multicast tree is constructed starting from the source of the data and connects all the destinations, i.e. there is only a single path between any source-destination pair. Such protocols are characterized by lower bandwidth consumption than their flooding counterparts. They suffer from low robustness when operating in highly mobile networks, since only a single path between a source-member pair is available. A tree-based protocol can be further categorized into the source-tree and shared-tree varieties. In source-tree protocols, the tree is rooted by the source node itself, whereas in shared-tree protocols, a single tree is shared by all multicast group sources and is rooted at a node known as the core node. The examples of source-tree protocols include the multicast zone routing protocol (MZRP) [17], multicast routing algorithms based on levy flying particle swarm optimization (LPSO) [18], TMRF [19] and multicast opportunistic cooperative routing in mobile ad-hoc networks (MO-CORMAN) [20].

Multicast ad-hoc on-demand distance vector routing protocol (MAODV) [6], shared-tree ad-hoc multicast protocol (STAMP) [21], reliable and energy-aware multicast ad-hoc on-demand distance vector (REA-MAODV) [4], cuckoo search and m-tree-based multicast ad-hoc on-demand distance vector (CS-MAODV) [22] and routing protocol for low-power and lossy networks (RPL) [23] are, in turn, instances of shared-tree protocols.

Mesh-based protocols allow data packets to be forwarded to the same receiver via different paths [24]. Numerous routes between the sender-receiver pair offer better protection against frequent topology changes and increase successful delivery rates [15]. However, the efficiency of mesh-based protocols is lower compared to tree-based protocols, due to multiple routes. Route discovery and mesh building are conducted using broadcasting to discover routes, or using core or central points for mesh building [15]. On-demand multicast routing protocol (ODMRP) [7], core assistant mesh protocol (CAMP) [25], and improved on-demand multicast routing protocol (IODMRP) [26] are examples of mesh-based protocols.

Hybrid multicast protocols combine both tree-based and mesh-based protocols in an attempt to achieve both performance and robustness [15], [16]. Similar to mesh-based approaches, multiple paths are constructed to forward data packets to their destinations. The tree-based approach is used in the route setup process to ensure multicast efficiency. Some examples of hybrid-based protocols include the following: ad-hoc multicast routing protocol (AMRoute) [27], efficient hybrid multicast routing protocol (EHMRP) [28], and zone-based energy aware hybrid multicast routing scheme (ZEHMRP) [29]. Hierarchical routing protocols aim to provide scalability and reduce the number of participating nodes by organizing them into a certain hierarchy. A group of nodes is used to form a cluster or a dominating set of nodes. The examples of cluster-based protocols include LACMQR [30] and EGMP [31]. Adaptive multicast routing protocols adjust their performance taking into account different environmental conditions. For example, the adaptive demand-driven multicast routing protocol (ADMR) [32] is capable of adjusting itself, taking into consideration the mobility state of the network. Once the network mobility level becomes very high, ADMR switches to flooding to overcome link breakages.

Location-based protocols assume that the locations of participating nodes are known. The geographical position of each node is determined using GPS receivers or other positioning services. Moreover, a location service is needed to obtain the positions of destination nodes. Racket forwarding is performed based on the information about the location of the direct neighbor nodes and of the intended destinations. So, nodes offering more efficient progress towards the destinations are selected, resulting in a reduced number of participating nodes. Since location-based multicast routing protocols scale well in large wireless networks, they have recently attracted researchers' attention. The properties of ad-hoc networks, such as constrained power and limited bandwidth, along with the need for scalable and energy efficient protocols, justify utilizing position-based routing in such networks [7]. However, multicasting deals with a group of members and carrying information about the positions of all multicast members in the packet header causes a scalability problem. The positions of a large set of destinations need to be maintained efficiently as well [9]. Some examples of location-based protocols include the following: scalable QoS multicast routing protocol (SQMRP) [33], position-based multicast routing protocol for ad-hoc network using backpressure restoration (PBMRP-BR) [34], scalable and predictive geographic multicast routing scheme in flying ad-hoc networks (SP-GMRF) [35], and location-aware multicasting protocol (LAMP) [36].

It has been observed that most of the existing protocols do not take the issue of scalability into consideration [9]. A crucial problem is that the control overhead may become high if the network is dense, large and/or includes a large number of destinations. Hence, in this research, the scalability and efficiency of multicast routing protocols have been considered.

In such a context, two popular and benchmark protocols have been proposed: MAODV and ODMRP. As the performance of most other protocols is compared to these [9], the rest of this section discusses both protocols in detail.

The MAODV routing protocol [6] uses the broadcast route-discovery approach to discover multicast routes on demand. Nodes participating in the network send a route request (RREQ) packet when they need to join a multicast group, or when they have data to send to a multicast group, and they do not have a route along such a packet could be sent. Only members of the projected multicast group are allowed to respond to a join RREQ. If the RREQ is not a join request, any node having a fresh route to this group can respond. Upon receiving a join RREQ to a group that it is not a member of, or upon receiving a RREQ to a group and not having a route thereto, an intermediate node rebroadcasts this RREQ to its neighbors.

Upon receiving a RREQ packet, the intermediate nodes update their route table. Nodes receiving a join RREQ for a specific multicast group are allowed to reply if they are members of the multicast group tree and the recorded multicast group's sequence number is at least as high as that included in the RREQ. Upon deciding to respond, a node updates its route and multicast route tables by placing the next hop information of the requesting node in the tables. Then, it unicasts a request response (RREP) back to the source node. Upon receiving the RREP, nodes along the path to the source create a forward path by adding a route table along with a multicast route table entry for the node that they received the RREP from.

The source node waits for a specific period of time and enables only the received route with the greatest sequence number and the lowest hop count to the nearest member of the multicast tree. Consequently, it enables the chosen next hop in its multicast route table, then unicasts an activation message (MACT) to the chosen next hop. The next hop, in turn, enables the source node entry in its multicast route table. If this node is a member of the multicast tree, it stops propagating this message. Else, it will have received one or more RREPs from its neighbors. It keeps the best next hop for its route, unicasts MACT to the selected next hop, and enables the correlated entry in its multicast route table. The aforementioned procedure continues until the RREP originating node has been reached. After that, data packets are forwarded only by nodes along the activated routes.

The first member joining the multicast group becomes the group leader. This leader maintains the multicast group sequence number and broadcasts it to the group members via a group hello message. The nodes use the group hello information for updating their request tables. Furthermore, MAODV has to actively track and react to changes in the tree resulting from membership changes and node movements.

ODMRP [7] uses a mesh-based approach. Hence, the multicast tree's drawbacks, such as alternating connectivity, frequent tree reconfiguration, and non-shortest path in a shared tree, are avoided [7]. In ODMRP, the multicast packets are forwarded only by a subset of nodes via scoped flooding. It conducts on-demand procedures to dynamically maintain multicast group membership and build routes. When a source has data to be sent and no already-chosen routes to the group members are available, the source broadcasts a join-query packet to the entire network. Join-query packets are broadcast periodically to update membership information and refresh the routes.

Backward learning for the reverse path back towards the source is used, i.e. routing tables are updated with the appropriate ID of the node from which the message was received. The message is rebroadcast if it is induplicate and TTL is larger than zero.

Upon receiving a join-query packet, a multicast receiver creates and broadcasts a join-reply to its neighbors. Once a node receives a join-reply, it checks if the next hop node ID of one of the entries is the same as its own ID. If so, the node realizes that it is a part of the forwarding group. Hence, it sets the FGFLAG. Accordingly, it broadcasts its join table built upon the matched entries. The next hop node ID field is filled by getting information from the nodes' routing tables. Thus, each forward group member propagates the join-reply until reaching the source via the designated shortest path.

After completing the route construction process and establishing the forwarding group, sources can multicast packets to the receiving nodes via these routes. While the source has data to be sent, it periodically sends join-query packets to keep the forwarding group and the routes fresh. A node forwards a data packet only if it is not a duplicate and the setting of the multicast group FGFLAG has not expired yet. This procedure reduces the traffic overhead and avoids sending packets over expired routes.

ODMRP adopts the soft state approach to maintain multicast group members. Hence, no explicit control packets are sent to leave a multicast group. When a source is about to leave the group, it simply stops sending join-query packets, as it no longer has data to be sent. If a receiver is no longer interested in a particular group, it does not send the join-reply for that group. Nodes in the forwarding group are treated as non-forwarding nodes if not refreshed before they timeout. The relaxed connectivity makes ODMRP more stable for mobile wireless networks [7].

3. Windmill Protocol

The proposed Windmill protocol consists primarily of five phases: network setup, network maintenance and membership update, route instantiation, route maintenance and, finally, data transmission. Table 1 presents the variables and notations used in further discussions.

Windmill assumes that NN cooperative nodes are distributed randomly in a square-shape area and are aware of their positions. During the network setup phase, the nodes collaborate to divide the area into zones and elect an initial ZL for each zone. After that, communication between ZL and the nodes interested in joining a specific group is conducted. In Table 2, the packets exchanged during the Windmill network setup phase are summarized. Upon using RDF, each node receiving a packet forwards the packet only if it is closer to the destination node than its previous

Table 1
Variables and notations used

Notation	Description	Notation	Description
NN	Number of nodes	ZL	Zone leaders
DS_n	Distance between the node and the center of its zone	DS_m	Maximum possible distance between a node and the center of a zone
SP_n	Node speed	SP_m	Maximum possible node movement speed
BT_n	Remaining battery life (time) of a node *n*	BTm	Maximum possible battery life (time)
CP_n	CPU processing power of a node *n*	CPm	Maximum CPU power available
MM_n	Memory usage of a node *n*	MM_m	Maximum memory capacity available
IP_n	IP address of node *n*	SN_n	Sequence number issued by node *n*
Pos_n	Position of node *n*	GID	Group number
$Z_{[x,y]}$	Zone number x, y	$ZL_{[x,y]}$	Zone leader of zone number x, y
D_{mov}	Movement distance allowed before sending PosUpdate	$PosZL_{[x,y]}$	Position of ZL of zone number x, y
DD	Distance between the forwarding node and the destination	D_{TH}	Number of destination nodes in a zone deciding to use RDF or ZBrd
RDF	Restricted directional flooding	ZBrd	Zone broadcast
$ProbL_{nz}$	Probability of node *n* being selected as ZL for its zone *z*	D_{cen}	ZL distance allowed from the zone center before sending ZLElect

Table 2
Packets sent during the network setup phase

Packet identifier	Stands for	Description
ZLProb	ZL probability	• Contains probability of a node to be elected as ZL of its zone • Sent from each node in a specific zone to nodes inside that zone, i.e. ZBrd
ZLPos	ZL position	• ZL of a zone to inform other nodes in it zone about its position • Sent using ZBrd
JoinGroup	Join group	• Nodes in a specific zone to ZL of that zone to inform it that they are interested in joining a specific group • Sent using RDF
PosUpdate	Position update	• Nodes in a specific zone to ZL of that zone to inform it about their position • Sent using RDF

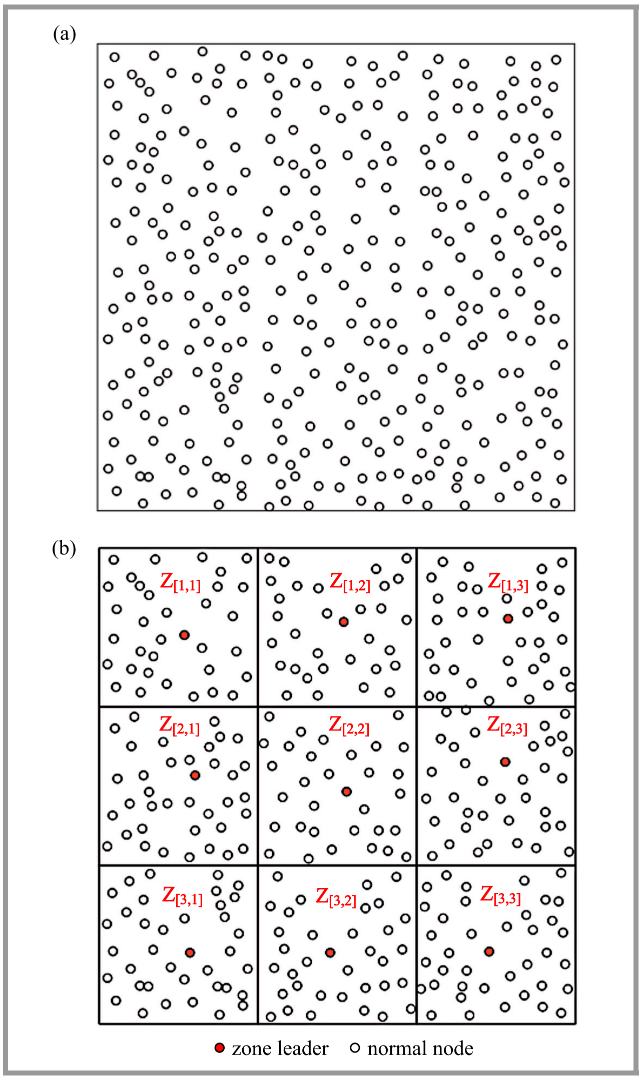

Fig. 1. Network structure at the beginning of the setup phase (a) and after the network setup phase, after division of the area and selection of ZLs (b).

hop. Upon using ZBrd, the nodes process a packet and forward it only if they are within the intended zone.

At the beginning of the network setup phase, the network's area is divided into numerous equal-size square-shape zones and initial ZLs for different zones are elected. Figure 1 shows the network structure at the beginning of the network setup phase, as well as after dividing the area into nine (3 × 3) zones and electing ZLs.

Each node knows the zone it belongs to using its position, the area coordinates, and the number of zones. Node position is known via GPS, while the area coordinates and the number of zones are stored in each node before deployment.

After dividing the area into zones, nodes inside each zone will start electing a ZL. The ZLs are chosen to be near the zone's center, in order to make sure that the time needed for communication between ZL and any node inside the zone is almost the same. Next, each node n inside a zone $Z_{[x,y]}$ is assigned a weight representing its probability of being the ZL of a particular zone. The most important aspects taken into consideration while selecting ZLs are the distance between the node and the center of the zone that the ZL will be responsible for $\mathrm{DS_n}$, the node's speed $\mathrm{SP_n}$ and battery remaining life time $\mathrm{BT_n}$. Choosing a ZL that is close to the center of the zone boundary and moving with a low speed increases the probability of the communication between ZLs of different zones being performed in one hop, which helps protect important packets. Choosing ZLs with low movement speeds also increases the probability that the elected ZL will stay in the zone longer, and so there is no need to re-elect a new ZL within a short period of time. Moreover, choosing a node with a high remaining battery life time reduces the likelihood the battery being drained, i.e. reduces the probability of electing a new ZL and transferring important and secure information in its possession.

Two other important factors that should be taken into consideration when electing a ZL are the CPU processing power $\mathrm{CP_n}$ and memory capacity $\mathrm{MM_n}$ of the nodes. ZLs with high CPU processing power and large memory significantly affect network performance, since these ZLs may be the bottleneck of the position management scheme.

Each node inside a specific zone uses these factors to calculate the probability of itself being elected as a ZL for a specific zone. Probability $\mathrm{ProbL_{nz}}$ of node n in zone z being elected as a ZL for that zone is:

$$\mathrm{ProbL_{nz}} = 0.2 \times \left(1 - \frac{\mathrm{DS_n}}{\mathrm{DSm}}\right) + 0.2 \times \left(1 - \frac{\mathrm{SP_n}}{\mathrm{SPm}}\right) + 0.2 \times \left(\frac{\mathrm{BT_n}}{\mathrm{BTm}}\right) + 0.2 \times \left(\frac{\mathrm{CP_n}}{\mathrm{CPm}}\right) + 0.2 \times \left(\frac{\mathrm{MM_n}}{\mathrm{MMm}}\right) , \quad (1)$$

where: $\mathrm{DS_m}$ is the maximum possible distance between a node and the center of a zone, $\mathrm{SP_m}$ is the maximum possible node movement speed, $\mathrm{BT_m}$ is the maximum possible battery life time, $\mathrm{CP_m}$ is the maximum CPU power available, $\mathrm{MM_m}$ is the maximum memory capacity available.

Values of the weights of different parameters are chosen equally, since we believe that they are all important when selecting the ZL. DSm is considered to be the distance between two opposite corners of a zone. $\mathrm{SP_m}$ is a predefined value that depends on the environment in which the protocol is deployed. $\mathrm{BT_m}$, $\mathrm{CP_m}$ and $\mathrm{MM_m}$ depend on the current technology found in the market.

After calculating its probability of being elected as a ZL, each node sends a ZLProb message to other nodes in its zone using zone broadcast ZBrd. Upon receiving the packet, each node will process it only if it is in the intended zone $\mathrm{Z_{[X,Y]}}$. Otherwise, the packet is dropped. The node with the highest probability in each zone will be the ZL of that zone. At this step, we assume that the network is error free and so all nodes within a specific zone receive the same set of ZLProb messages. Now, the ZL node sends the ZLPos message to inform other nodes in its zone about its position. This message is also sent using ZBrd.

After that, only the interested nodes send JoinGroup and PosUpdate messages to the ZL of their zone to inform it that they are interested in joining a specific group and to tell it about their positions. These packets are sent via

Table 3
Packets sent during the network maintenance and membership update phase

Packet identifier	Stands for	Description
ZLElect	ZL election	• Sent from ZL of a specific zone to nodes inside that zone to initiate a new ZL election process • Sent using ZBrd
ZLQuery	ZL query	• Sent by a node entering a new zone to ask about its ZL • Sent to first hop neighbors
LeaveGroup	Leave group	• Nodes in a specific zone to ZL of that zone to inform it that they are no longer interested in a specific group or when they are leaving the zone • Sent using RDF
ZLProb, ZLPos, JoinGroup, and PosUpdate		• As explained in the network setup phase

Table 4
Packets sent during the route discovery phase

Packet identifier	Stands for	Description
SRREQ	Source route request	• Request sent from the source node using RDF to local ZL to ask about destination nodes for the multicast session to be held • Sent using ZBrd
IRREQ	Internal route request	• Request sent from a specific ZL to the interested local destinations to join the multicast session held • This packet is sent using ZBrd if the number of destination nodes in this zone is greater than D_{TH}, else it is sent using RDF towards each destination
ERREQ	External route request	• Request sent from ZL of a given zone to neighbor ZLs using RDF, to ask about destination nodes in the neighbor zone that are interested in joining the multicast session

RDF. The use of RDF offers a high probability of finding a path compared to the greedy solution. Such an approach also reduces the resulting overhead compared with blind broadcasting to the entire network.

3.1. Network Maintenance and Membership Update

During the network lifetime, nodes may move freely within the network, may move in and out of the network and change their group membership. The proposed protocol tries to cope with these issues. In Table 3, the packets exchanged during the network maintenance and membership update phase of Windmill are summarized.

Let us start with non-ZL nodes. Members joining a specific group can leave it by sending a LeaveGroup packet to the ZL of their zone. Moreover, any node can send JoinGroup and PosUpdate messages to its zone ZL if it becomes interested in a specific group. These packets are sent via RDF and contain the same fields as described in the network setup phase.

Member nodes should also inform their ZLs about their new position if they have moved a predefined distance D_{mov} from their last known position. When a specific member is about to leave the boundaries of its zone, it should send a LeaveGroup message to the previous ZL. Then, it sends a ZLQuery packet to ask about the ZL of the new zone. This packet is sent to first hop neighbors and any node in new zone may reply by sending ZLPos packet containing the IP and position of the responsible ZL. Now the moving node can communicate with the new ZL by sending JoinGroup and PosUpdate messages.

Regarding ZL nodes, a ZL sends a ZLPos message to inform other nodes in its zone about its new position if it has moved D_{mov} from its last known position. This message is sent using ZBrd.

If the ZL decided to depart its zone, its distance from the zone center became higher than a pre-defined distance D_{cen}, or if its battery is about to turn off, it may send a ZLElect packet to initiate a new ZL election. This packet is sent using ZBrd. Upon receiving this packet, each node inside the zone will calculate its probability to become a ZL and a new ZL will be elected, as discussed in the network setup phase.

3.2. Discovery Phase

Table 4 presents the control packets exchanged to handle the route discovery algorithm. When a source node decides to initiate a multicast session, a source route request (SRREQ) packet is first directed to its local ZL node to ask for possible participating nodes in the multicast session held. The SRREQ packet continues to be propagated restrictedly using RDF, until it reaches the intended ZL.

When the source ZL node receives the SRREQ packet, it sends an external route request (ERREQ) packet to the four neighbor ZLs. Here, we consider the case that ZLs of adjacent zones may not be within the transmission range of each other. Hence, multi-hop routing is assumed and packets are sent across zones using RDF.

The source ZL also sends an internal route request (IRREQ) packet only if there are interested nodes within this zone. This packet is sent trying to find routes to the participating nodes within this zone. Upon receiving the packet, each node will process it only if it is in the intended zone $Z_{[X,Y]}$. Otherwise, the packet is dropped. This packet is sent using ZBrd if the number of destination nodes in this zone is greater than D_{TH}. In the zone broadcast, upon deciding to forward the packet, the node stores the IP address of its previous hop IPI to be used in the reverse path. Also, it alters the IPI field to be its own IP address and proceeds with forwarding the packet. On the other hand, if the number of destination nodes in this zone is lower than or equal to D_{TH}, RDF will be used. In this case, $ZL_{[X,Y]}$ will prepare a separate packet for each destination, and each node processing the packet will forward it only if it is closer to that destination.

Upon receiving ERREQ for the first time, the intended neighboring ZL continues the route discovery process by finding a route between itself and the neighbor ZLs (by sending ERREQ), and later between itself and other destinations in its zone (by sending IRREQ). The ERREQ packet is propagated until it reaches all the network zones using the forwarding strategy, as discussed later on.

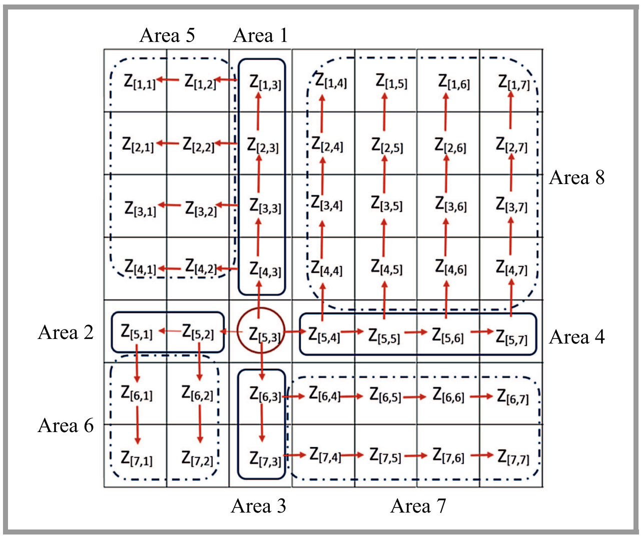

Fig. 2. Forwarding ERREQ packet in Windmill protocol.

The proposed protocol utilizes the network division to forward the ERREQ packets to discover the anticipated group members with very low overhead, as well as to prevent sending duplicate packets. In this subsection, the forwarding of ERREQ packets between the network zones is explained – see Fig. 2. The decision to forward the ERREQ packet to the neighbor zones is the responsibility of the ZL node.

The source node resides in zone $Z_{[5,3]}$. Firstly, the ERREQ packet is forwarded towards the border of the four neighbor zones as the first forwarding step (in our example, there are zones $Z_{[4,3]}$, $Z_{[5,2]}$, $Z_{[6,3]}$ and $Z_{[5,4]}$ are present).

If each zone receiving the ERREQ packet resends it to all 4 of its neighbors, meaning that a lot of duplicate packets are produced. To overcome this, an efficient forwarding strategy is proposed. This algorithm enables the ZL of each zone to take part in delivering the packet to two neighbor zones at the most. In this forwarding scheme, the ZL is based on the number of the source zone $Z_{[X,Y]}$, and the coordinates of the intermediate zone that is currently forwarding the packet $Z_{[x,y]}$. This forwarding strategy ensures that the ERREQ packet is propagated through the network with no duplicates and all the network zones are visited only once (see to Fig. 2).

For example, assume that the packet is sent out from zone $Z_{[5,3]}$. Here, the ZL node of zones $Z_{[4,3]}$, $Z_{[3,3]}$, $Z_{[2,3]}$ and $Z_{[1,3]}$ (area 1) forwards the packet to the zones that are above and to the left of the current zone (if any). In the following step, zones $Z_{[1,2]}$, $Z_{[2,2]}$, $Z_{[3,2]}$, $Z_{[4,2]}$, $Z_{[1,1]}$, $Z_{[2,1]}$, $Z_{[3,1]}$ and $Z_{[4,1]}$ (area 5) send the packet only towards zones to their left (if any). A similar strategy is used for packets forwarding to other network parts to eliminate duplicate packets.

The pseudocode of the forwarding strategy is illustrated below, considering that the source zone is $Z_{[X,Y]}$, and the current zone to forward the packet is zone $Z_{[x,y]}$:

- if $x = X$ and $y = Y$ (source zone), then forward to zones $Z_{[X-1,Y]}$, $Z_{[X,Y-1]}$, $Z_{[X+1,Y]}$ and $Z_{[X,Y+1]}$,
- if $y = Y$ and $x < X$ (area 1), then forward to zones $Z_{[X,Y-1]}$ and $Z_{[X-1,Y]}$,
- if $x = X$ and $y < Y$ (area 2), then forward to zones $Z_{[X,Y-1]}$ and $Z_{[X+1,Y]}$,
- if $y = Y$ and $x > X$ (area 3), then forward to zones $Z_{[X,Y+1]}$ and $Z_{[X+1,Y]}$,
- if $x = X$ and $y > Y$ (area 4), then forward to zones $Z_{[X,Y+1]}$ and $Z_{[X-1,Y]}$,
- if $x < X$ and $y < Y$ (area 5), then forward to zone $Z_{[X,Y-1]}$,
- if $x > X$ and $y < Y$ (area 6), then forward to zone $Z_{[X+1,Y]}$,
- if $x > X$ and $y > Y$ (area 7), then forward to zone $Z_{[X,Y+1]}$,
- if $x < X$ and $y > Y$ (area 8), then forward to zone $Z_{[X-1,Y]}$.

Figure 3 shows the control packets exchanged during the route discovery phase of the Windmill protocol.

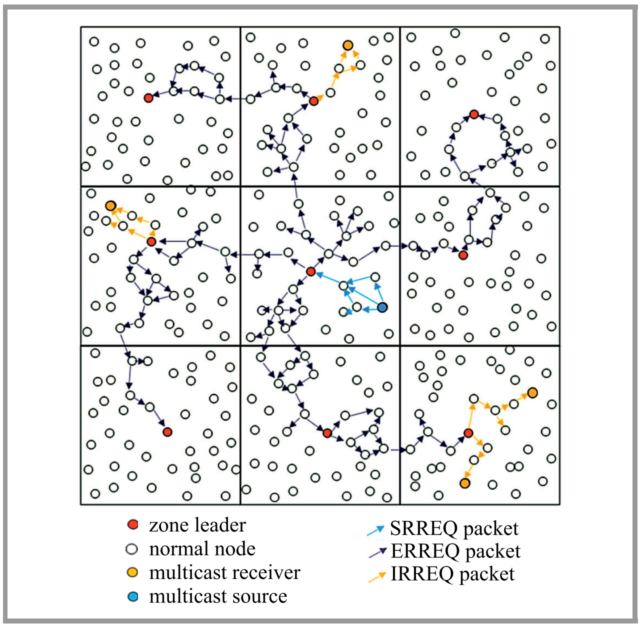

Fig. 3. Packets sent during the route discovery phase.

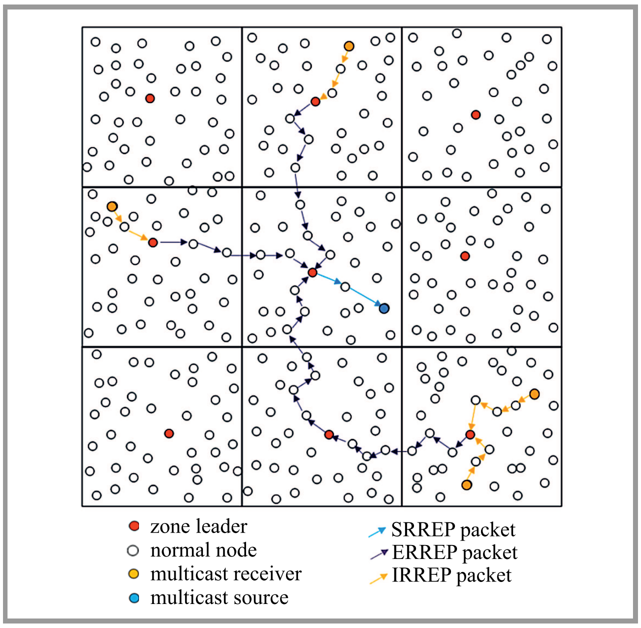

Fig. 4. Packets sent during the route setup phase.

3.3. Route Setup

The next step, after propagating the request packets, is to setup the routes by sending the reply packets. Table 5 contains the control packets exchanged to handle the route setup phase.

After forwarding the IRREQ packet and if it is interested in participating in the session, node J commences the process of setting up a route from the local ZL to itself by sending an internal route reply (IRREP) packet. Each intermediate node forwards this packet to the node from which it received the corresponding IRREQ packet. This process continues until the packet reaches the intended ZL.

To reduce the network overhead, each zone leader $\mathrm{ZL}_{[x,y]}$ sends only one external route reply (ERREP) to the neighbor ZL that forwarded the original ERREQ to it. This packet is sent using the reverse path, until the ZL node that issued the original ERREQ packet is reached.

To further reduce the overhead in the network, the source zone leader $\mathrm{ZL}_{[X,Y]}$ sends only one source route reply (SRREP) to the source node S. Each node sends this packet to the previous hop from which it received the original SRREQ packet, until the packet reaches node S.

Figure 4 shows the control packets exchanged during the route setup phase.

3.4. Route Maintenance

During data transmission, some nodes may not receive data packets due to broken links caused by failure or movement of the nodes. When a link break is detected, the node located upstream of the broken link sends a route error (RERR) packet backwards to the upstream nodes to inform them about this failure. Intermediate upstream nodes, upon receiving this packet, clear the information related to the downstream nodes, and re-forward the packet towards their upstream nodes. Also, the nodes located downstream of the broken link will clear the related entries and free the resources when a predefined time has elapsed without receiving data from the upstream nodes.

When a ZL receives the RERR packet, it deletes the related entry from its routing table and initiates a new route discovery process towards the affected destinations. Also, if the source receives a RERR packet, it discovers that the link between itself and the local zone leader is no longer

Table 5
Packets sent during the route setup phase

Packet identifier	Stands for	Description
SRREP	Source route reply	• Reply sent from ZL of the zone of the source node indicating that there are nodes in the source zone want to join the multicast session held
IRREP	Internal route reply	• Reply from a given node to its local ZL setting up a route to itself • Nodes reply to the first IRREQ they receive
ERREP	External route reply	• Reply sent from ZL of a given zone to the ZL of the zone from which it received the ERREQ packet. This packet indicates that there should be a route passing through this ZL

Table 6
Packets sent during the route maintenance phase

Packet identifier	Stands for	Description
RERR	Route error	When a broken link is encountered during data transmission, the node that discovers the broken link informs its upstream nodes about this failure using RERR packet

available. Accordingly, the source node deletes the related entry from its routing table and initiates a new route discovery process to reconstruct the broken route towards the local ZL. Table 6 shows the control packet exchanged to ensure route maintenance.

3.5. Data Transmission

The source node waits for a predefined time to setup the routes to the nodes that want to participate in the multicast session. Then, it starts sending data packets to the multicast group members using the chosen routes. The multicast data packets are sent along the multicast tree, from the source to the ZL nodes. Whenever a data packet reaches the ZL nodes, the ZL nodes forward a copy of the received data packet to the members in their zone. Each intermediate node simply re-forwards data packets to its successor in the route determined during the route initiation process.

4. Performance Evaluation

In this section, a simulated performance evaluation of MAODV, ODMRP and Windmill is presented. MAODV and ODMRP protocols are considered for comparison purposes, since they were proposed by the mobile ad-hoc networks working group at the IETF and are often considered as benchmarks for evaluating performance of ad-hoc multicast routing protocols [8].

Global Mobile Simulation (GloMoSim) [37] is used as a simulation tool to evaluate the performance of the three protocols under consideration. A network with 60 mobile nodes located within an area of 1000 m × 1000 m that is divided into 4×4 zones is considered. The nodes' transmission range of 250 m and channel capacity of 2 Mbit/s are used. The initial positions of the nodes are chosen randomly. After that all nodes are allowed to move in accordance with the random waypoint mobility principle, i.e. each node travels to a randomly selected location at a configured speed and then pauses for a configured pause time, before choosing another random location and repeating the same steps. A pause time between 0 and 10 s is simulated. The maximum node mobility speed is 40 km/h.

The 802.11 MAC layer and constant bit rate (CBR) traffic over user datagram protocol (UDP) have been used. For either protocol, a routing packet processing delay of 1 ms is assumed. In order to minimize collisions, a random delay between 0 and 10 ms is introduced before retransmitting the broadcast packets. Sources and destinations are chosen randomly. One multicast group with a single source and 20 members is simulated. The source sends data at the rate of 20 packets/s. The size of data payload is 512 bytes. Multicast group members are allowed to join and leave the multicast group at any time during the simulation. Member nodes are selected randomly with uniform probabilities. Each simulation is performed for 300 s.

4.1. Performance Metrics

Five important parameters related to ad-hoc network multicast transmissions have been tested. These parameters include the following: node mobility speed, number of sources, multicast group size (members), network size and number of zones. For each parameter, five performance metrics are evaluated. The metrics were derived from the ones suggested by the IETF mobile ad-hoc network working group for the purpose of evaluating routing/multicast protocols [38]:

1. **Packet delivery fraction (PDF)**. The ratio of the number of data packets actually really delivered to the multicast receivers versus the number of data packets supposed to reach them. This evaluates the protocol's ability to discover and maintain routes, as well as its effectiveness in delivering data to the intended receivers.
2. **Number of control packets transmitted per data packet delivered (CPD)**. Instead of using a pure control overhead, we choose to use a ratio of control packets transmitted to data packets delivered in order to investigate how efficiently control packets are utilized in delivering data to the intended receivers. Packets used for route instantiation and maintenance are considered upon calculating this metric. Furthermore, packets sent to construct and maintain the network's structure, update node positions and maintain membership are considered as well. The transmission at each hop along the paths is included in the calculation of this metric.
3. **Number of control and data packets transmitted per data packet delivered (CDPD)**. This metric shows the efficiency in terms of channel access and is very important in ad-hoc networks, since link layer protocols are typically contention-based.
4. **Average path length (APL) [hop]**. The average length of the paths discovered by the protocol. It is calculated by taking the average number of hops taken by each data packet to reach the destination.
5. **Average route latency (ARL) [ms]**. The average delay needed for discovering a route to the destination. It is defined as the average delay between sending

a route request/discovery packet by a source and receiving the first corresponding route reply packet. If a request is timed out and requires to be retransmitted, the sending time of the first transmission is used in calculating the latency.

Each point in the following figures is obtained by averaging the results of five simulation runs with similar configurations but various, randomly generated numbers.

4.2. Node Mobility Speed Effect

The node mobility speed has been varied to evaluate the ability of the protocols to deal with route changes. Figure 5a shows the PDF of the three protocols as a function of mobility speed. ODMRP is more effective than AODV and Windmill in PDF, as the maximum node speed is increased from 0 to 80 km/h. This is caused by ODMRP mesh topology which allows for alternative paths and makes ODMRP more robust compared to MAODV and Windmill which rely on a single path in their multicast tree. PDF for the three protocols decreases with increasing mobility speed, due to higher probability of link breakages and data packet drops.

Since most ad-hoc network medium access control protocols are contention based, having less packets transmitted per data packet delivered is very important [7]. As shown in Fig. 5b-c, CPD and CDPD for ODMRP are higher than those for MAODV and Windmill. The increased ODMRP

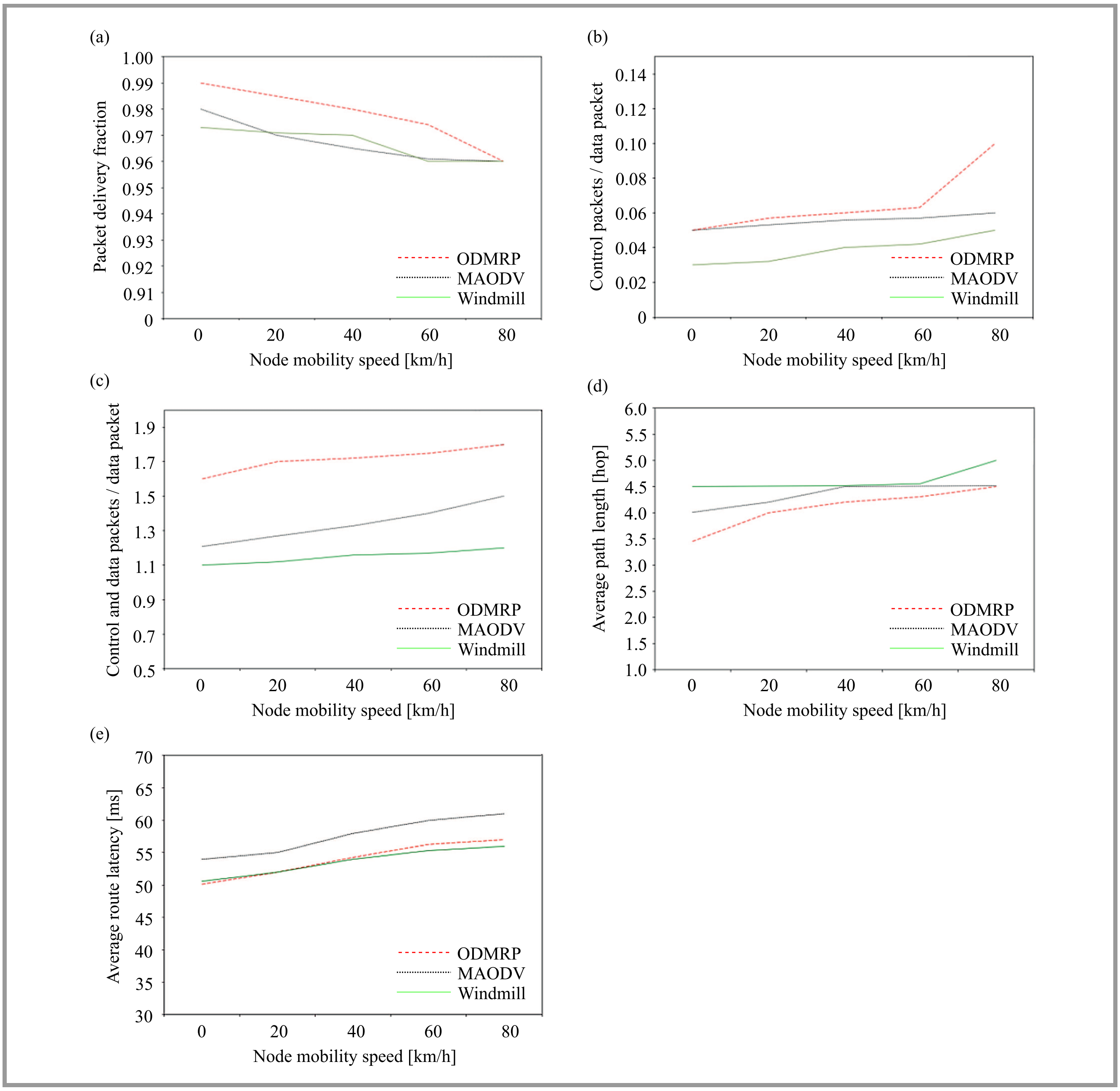

Fig. 5. Node mobility speed simulations.

CPD is due to its broadcast of the route request and reply and the periodic refresh of routes from source to different destinations. Moreover, bidirectional trees used in MAODV and Windmill are more efficient, compared to mesh, and avoid sending numerous copies of data packets to receivers, i.e. lower MAODV and Windmill CDPD.

In MAODV, the multicast group leader maintains updated multicast tree information by sending periodic group hello messages. Windmill, on the contrary, does not require sending periodic group hello messages. Moreover, MAODV sends the request packet to the entire network, whereas in Windmill, the request packets between zones are sent using RDF, and RDF or ZBrd are used only inside zones having destinations inside them. These two points justify the lower value of CPD of Windmill compared to MAODV. As far as network structure maintenance is concerned, in Windmill, the process of dividing the area into zones and initial ZL election is conducted once, at the beginning of the network setup phase. After that, any updates such those concerning nodes joining and leaving groups, position updates and new ZL election processes, are performed locally, inside the intended zone and most properly using RDF. Hence, the impact of network structure maintenance group membership on the control overhead is not noticeable. CPD and CDPD for the three protocols slightly increase with an increase in mobility speed, due to higher probability of link breakages and route repairs.

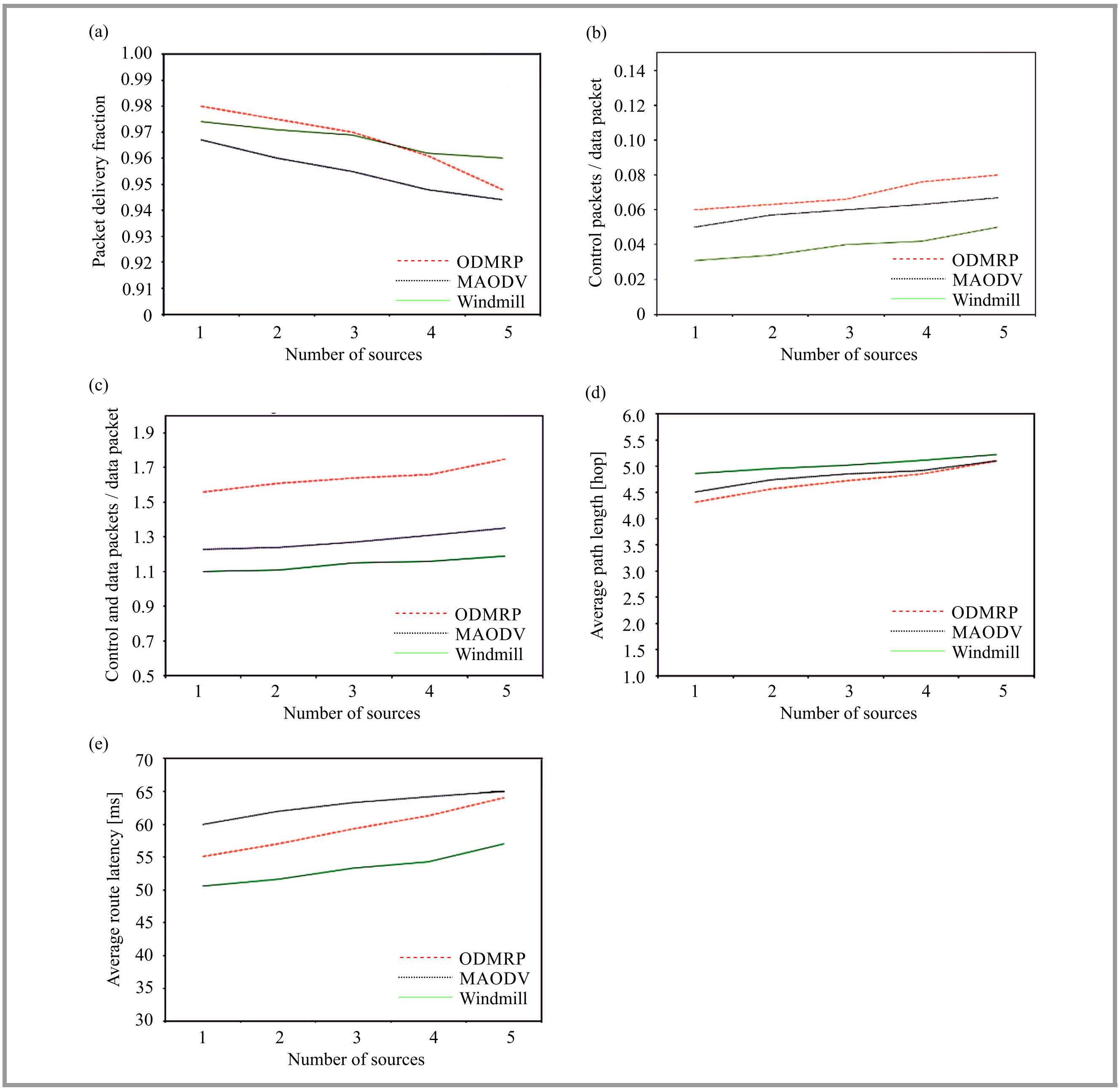

Fig. 6. Number of sources.

Regarding APL of the selected routes, Fig. 5d shows that routes in Windmill are a little bit longer than those in MAODV and ODMRP, since the routes are forced to pass through ZLs.

Figure 5e demonstrates that ARL of MAODV is higher than in the case of two other protocols. MAODV does not activate a multicast route immediately. A potential multicast receiver waits for a specified period of time, allowing to receiving numerous replies before sending an activation message along the chosen multicast route. On the contrary, ODMRP and Windmill activate the routes immediately. Moreover, ARL of the proposed protocol is a little bit lower than that of ODMRP, since the number of request and reply packets received by each node in Windmill is lower, which reduces the time spent by these nodes on processing these packets.

APL and ARL for the three protocols slightly increase along with increasing mobility speed, due to the higher probability of link breakages and choosing other, longer routes.

4.3. Number of Sources Effect

Next, the number of senders in the multicast group has been varied in order to evaluate the scalability of different protocols with respect to source nodes and the resulting effective traffic load. Figure 6a presents the PDF of the

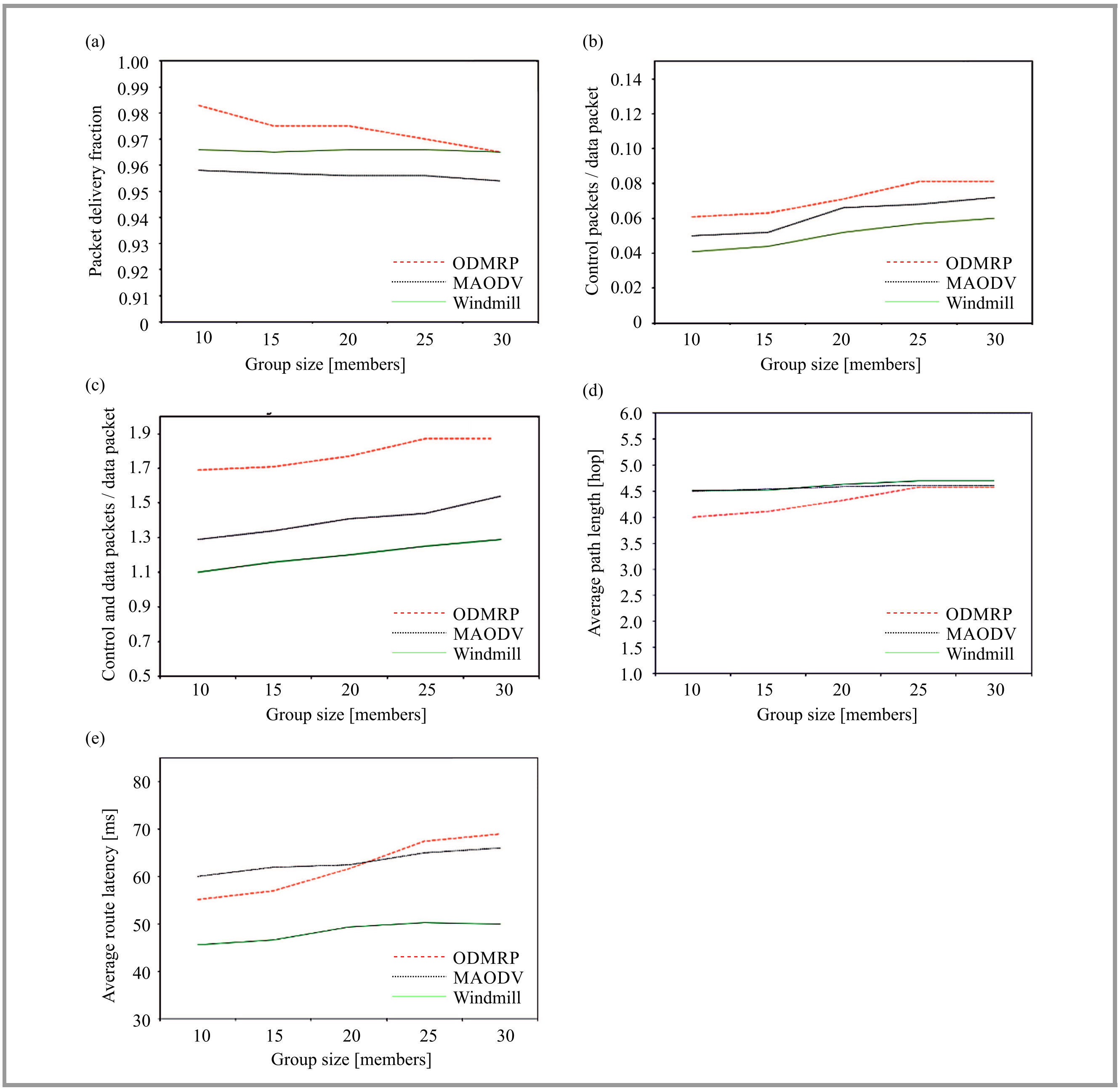

Fig. 7. Simulation results with varying group sizes.

three protocols as a function of the number of senders. ODMRP is more effective compared to MAODV in PDF when the number of senders is low due to its mesh topology. However, upon increasing number of senders from 1 to 5, ODMRP in particular does not scale well for PDF. In ODMRP, every source node sends out, periodically route requests through the network. When the number of source nodes becomes larger, this causes congestion in the network and the PDF drops significantly. MAODV, on the other hand, maintains only one multicast group leader that periodically sends group hellos through the network. Therefore, MAODV is more scalable compared to ODMRP [8]. As far as the proposed protocol is concerned, no periodic packets are sent. However, the network structure will be constructed for each source, as this justifies the moderate decrease of PDF in Windmill.

Figures 6b-c show that CPD and CDPD for the three protocols increase slightly with the increasing number of senders, due to congestion resulting from packets being sent periodically in ODMRP and MAODV, and from the network structure of Windmill. This congestion also justifies the increase in ARL shown in Fig. 6e.

Regarding APL of the selected routes, Fig. 6d shows that in Windmill are a little bit longer than those in both MAODV and ODMRP, since the routes are forced to pass through ZLs. However, this difference decreases as the number of sources increases, due to congestion forcing MAODV and ODMRP to choose longer paths.

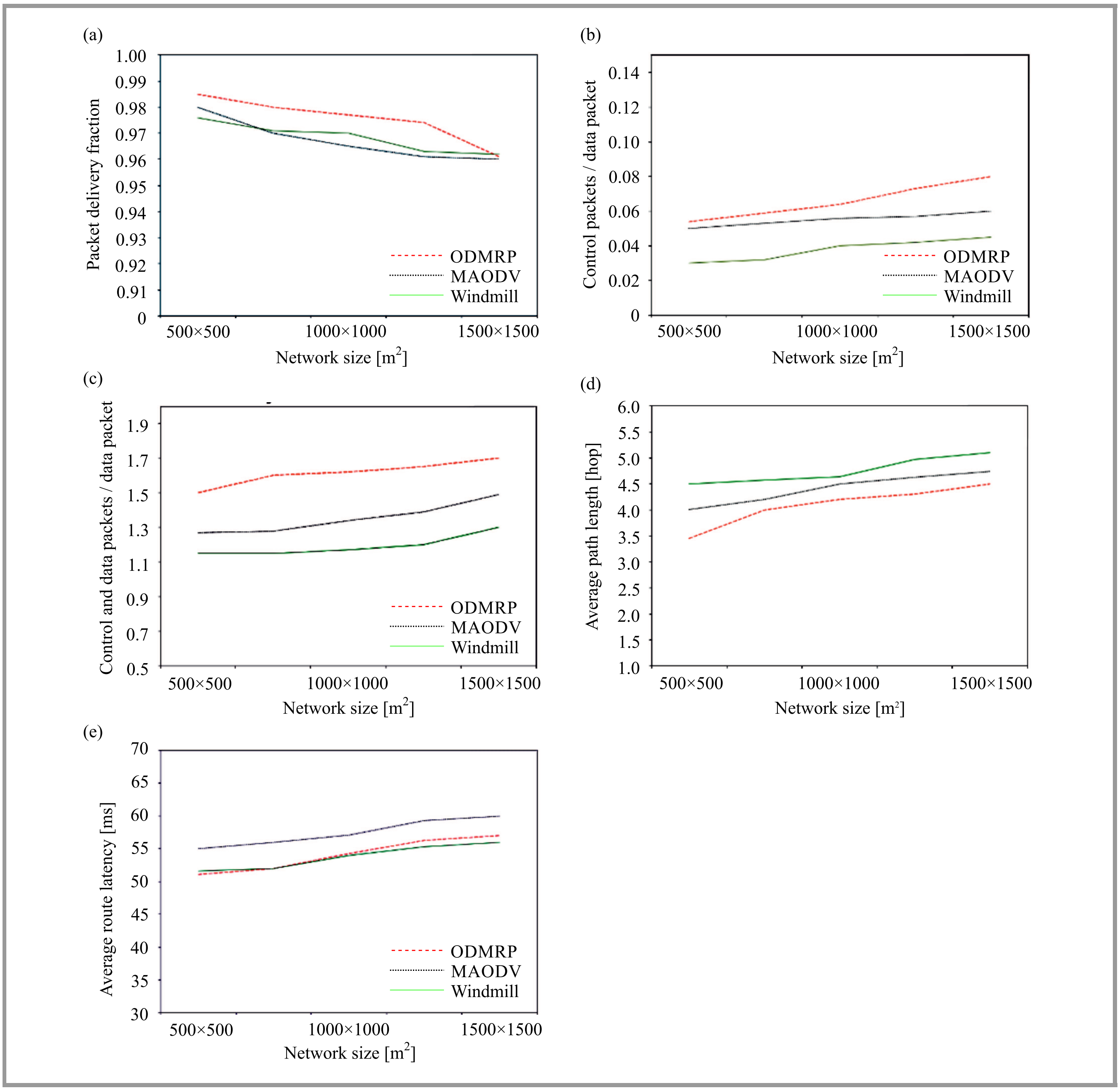

Fig. 8. Simulations with a varying network size.

4.4. Multicast Group Size Effect

In this scenario, the size of the multicast group is varied to examine the scalability of the protocol regarding number of members. Figure 7a shows that ODMRP is more effective than MAODV regarding PDF when the number of multicast group members is low. However, ODMRP does not scale well with multicast group size. There is a noticeable decline in PDF as the multicast group increases to 30 members. This can be attributed to collisions that occur from the frequent broadcasts through the network [8].

MAODV and Windmill scale better in terms of CPD and CDPD compared to ODMRP (Figs. 7b–c). This is due to ODMRP broadcast of route request and reply packets, periodic refresh of routes from the source to different destinations, and sending multiple copies of data packets to receivers. This increased number of packets also contributes to an increase in ARL and APL due to the time spent by participating nodes on processing these packets and the increased copies of data packets passing through longer paths.

4.5. Network Size Effect

For the fourth set of simulations, we varied the network size in order to evaluate the protocols' scalability for larger network areas. Different network sizes have been considered with node density of 60 nodes/km^2. Hence, the studied networks are 500 ×500 m with 15 nodes, 750 ×750 m with

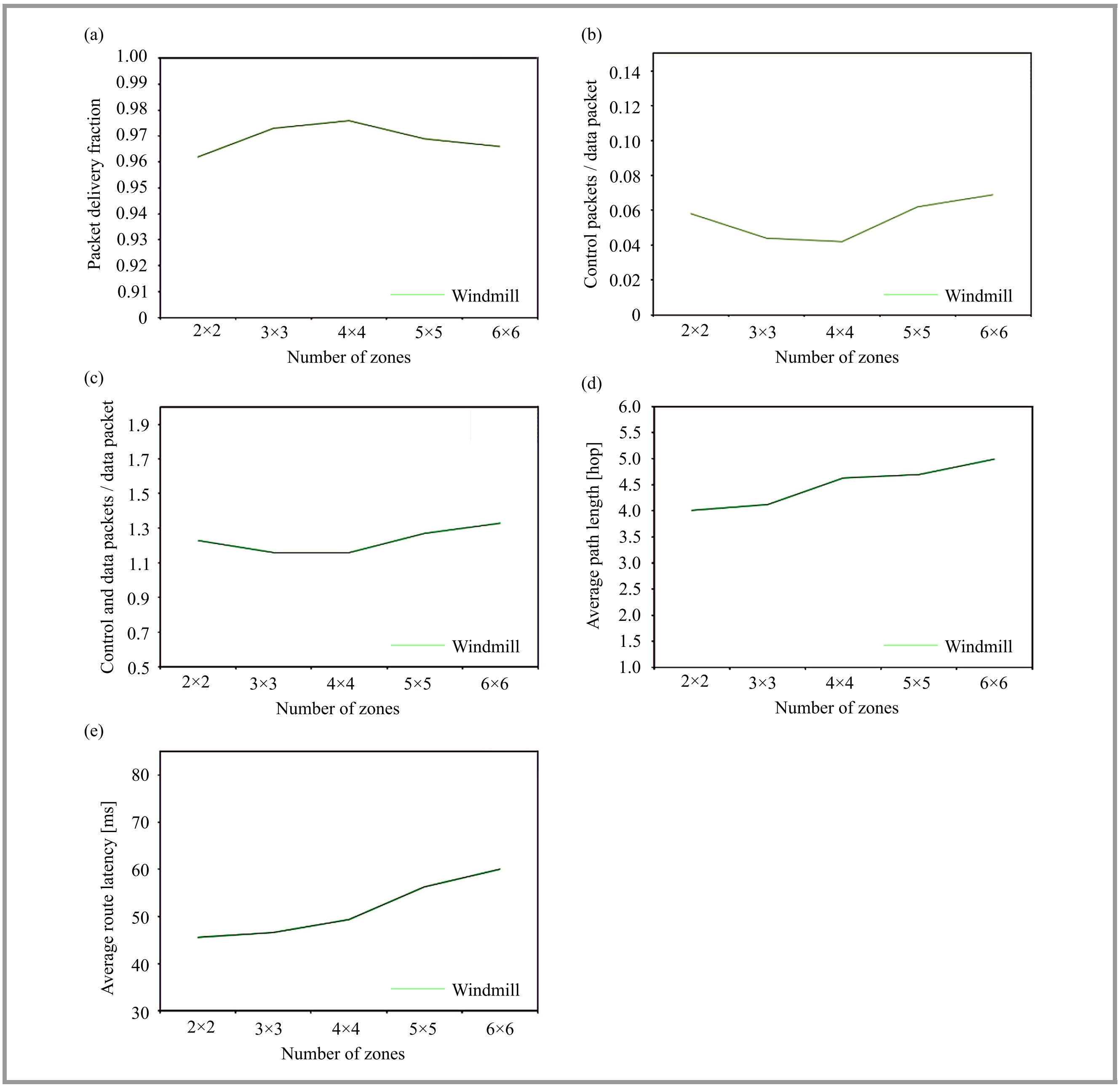

Fig. 9. Simulations results with varying number of zones.

34 nodes, 1000×1000 m with 60 nodes, 1250×1250 m with 94 nodes, and 1500×1500 m with 135 nodes.

Figure 8a shows that PDF of the three protocols decreases with an increase in network size. Larger network size increases the probability of having the source and destination nodes far away from each other, which means that longer routes are established and a higher probability of link breakages and data packet drops exists.

Figures 8b–c reveal that CPD and CDPD for the three protocols increase with an increase in network size due to longer routes and higher probability of link breakages that require route repairs and high control over the packets. CPD and CDPD for ODMRP are still higher than those for MAODV and Windmill, due to increased control and data packets.

As far as APL of the selected routes is concerned, Fig. 8d shows that routes in the proposed protocol are still a little longer than in MAODV and ODMRP, due to the routes passing through ZLs. APL and ARL for the three protocols increase with an increase in the network size, due to increased probability of having the source and destination nodes far away from each other, i.e. longer routes and extended setup times.

4.6. Number of Zones Effect

This parameter has been studied only for the Windmill protocol, since it is the only protocol dealing with the network as zones. To examine the effect of the number of zones, a network of 1×1 km is considered. This network is divided into 4 zones, each having the dimensions of 500×500 m, 9 zones – each of 333.33×333.33 m, 16 zones each of 250×250 m, 25 zones each of 200×200 m and, finally, 36 zones – each of 166.67×166.67 m.

Figure 9a shows that Windmill's PDF is always above 96%. This is an indication that it is highly effective in discovering and maintaining routes, regardless of zone size. Nevertheless, the highest PDF is obtained upon dividing the network into 9 and 16 zones.

Figures 9b–c reveal that the minimum CPD and CDPD values are obtained upon dividing the network into 9 and 16 zones. A large number of zones, i.e. with a small zone size, results in a higher probability of nodes moving from one zone to another, which means a higher control overhead required to maintain the network structure and group membership information, as well as to maintain the routes. A large number of zones also means a higher control overhead needed to discover external routes. On the other hand, an increase in zone size results in a higher probability of having destination nodes in each zone, which means a higher control overhead required to discover internal routes, especially when using ZBrd.

APL and ARL increase along with the increasing number of zones (Figs. 9d–e). A large number of zones means longer routes due to forcing the routes to pass through ZLs.

The analysis shows better performance in terms of PDF, CPD and CDPD for Windmill, when the network is divided into 9 and 16 zones. Moreover, moderate performance in terms of APL and ARL is achieved upon dividing the area into 9 or 16 zones. Hence, it is recommended to divide the network into 3×3 or 4×4 zones.

5. Results Summary and Discussion

Numerous conclusions may be drawn from the simulation results presented in the previous section:

- PDF for the three protocols is above 95% in most scenarios. This indicates that the three protocols are effective in discovering and maintaining routes for data delivery, even with fairly high node mobility levels and large area networks.
- The proposed protocol performs well in terms of scalability, as it maintains the minimum CPD and CDPD levels in all scenarios. The main reason behind the gap between CPD and CDPD levels typical of Windmill and those of MAODV and ODMRP is that nodes in MAODV and ODMRP are unaware of their and other nodes' positions. Hence, all request packets are sent using broadcasts to the entire network. Additionally, both protocols require sending periodic messages. Windmill, however, does not rely on sending periodic messages. Furthermore, request packets are sent between zones using RDF, with RDF or ZBrd being only used inside zones, with destinations located inside them.
- Slightly longer routes (higher APL) compared to MAODV and ODMRP are the only expense of using the new protocol, since routes in Windmill are forced to go through ZLs.
- Roughly speaking, an increase in node mobility speed, number of sources, multicast group size, and network size results in decreasing PDF and increasing CPD, CDPD, APL and ARL for the three protocols. This is mainly due to higher probability of link breakages which require route repairs and numerous control packets.
- When using Windmill, it is recommended to divide the network into 3×3 or 4×4 zones, since better performance in terms of PDF, CPD and CDPD, as well as moderate performance in terms of APL and ARL is achieved when dividing the area into 9 or 16 zones.

6. Conclusions and Future Works

The establishment of efficient routes between sources and the anticipated destinations is an important issue in mobile ad-hoc networks. This paper proposes Windmill, a hierarchal multicast routing protocol that seeks to enhance performance and scalability by dividing the network into zones and by relying on RDF. The novel protocol has been assessed and compared with its MAODV and ODMRP counterparts. In MAODV and ODMRP, the nodes are unaware of their and other nodes' positions. Hence, all request packets are sent using a broadcast to the entire network. Additionally, both protocols require sending periodic messages. Windmill, however, does not involve sending periodic messages. Furthermore, the request packets are sent between the zones using RDF, with RDF or ZBrd being only used inside the zones, with destinations located inside them.

A detailed performance evaluation has been conducted. Simulation results illustrate the efficiency of the three protocols in discovering and maintaining routes. Moreover, Windmill performs well in terms of scalability, as it maintains the minimum CPD and CDPD levels, even with high node mobility levels, large number of sources, large multicast groups, and large networks. Windmill's reduced CPD and CDPD levels are a consequence of using restricted directional flooding to send request packets. On the other hand, the proposed protocol's reduced overhead comes at the price of slightly longer routes.

There are still many open research issues related to ad-hoc networks, such as quality of service and energy efficiency. Security-related aspects stemming from the existence of malicious nodes performing different types of attacks are an interesting area of research as well. This work considered nodes that were evenly distributed from the geographical point of view. So, it is one of our future tasks to study scenarios with dense and sparsely populated regions of the network. Moreover, some improvements may be introduced to Windmill as well, such as turning into a dynamic/adaptive protocol by changing some details concerning the current state of the network. Lastly, we aim to implement and test the proposed protocol in real world conditions.

References

[1] A. Goyal, V. Sharma, S. Kumar, and K. Kumar, "Modified local link failure recovery multicast routing protocol for MANET", *J. of Inform. and Optimiz. Sci.*, vol. 41, no. 2, pp. 669–677, 2020 (DOI: 10.1080/02522667.2020.1733202).

[2] S. Ghasemnezhad and A. Ghaffari, "Fuzzy logic based reliable and real time routing protocol for mobile ad hoc networks", *Wirel. Personal Commun.*, vol. 98, pp. 593–611, 2018 (DOI: 10.1007/s11277-017-4885-9).

[3] M. Qabajeh and L. Qabajeh, "A survey and comparative study for performance evaluation technologies for wireless networks", *Int. J. of Comp. Sci. and Netw. Secur.*, vol. 19 no. 10, pp. 43–50, 2019 [Online]. Available: http://paper.ijcsns.org/07_book/201910/20191008.pdf

[4] A. Tavizi and A. Ghaffari, "Tree-based reliable and energy-aware multicast routing protocol for mobile ad hoc networks", *The J. of Supercomput.*, vol. 74, no. 11, pp. 6310–633, 2018 (DOI: 10.1007/s11227-018-2562-8).

[5] L. Qabajeh and M. Qabajeh, "Detailed performance evaluation of ARANz, ARAN and AODV protocols", *J. of Theoret. and Applied Inform. Technol.*, vol. 98. no. 12, pp. 2109–2131, 2020 [Online]. Available: http://www.jatit.org/volumes/Vol98No12/10Vol98No12.pdf

[6] E. Royer and C. Perkins, "Multicast operation of the ad hoc on demand distance vector routing protocol", in *Proc. 5th Ann. ACM/IEEE Int. Conf. on Mob. Comput. and Network. MOBICOM 1999*, Seattle, VA, USA, 1999, pp. 207–218 (DOI: 10.1145/313451.313538).

[7] S. Lee, M. Gerla, and C. Chiang, "On-demand multicast routing protocol", *IEEE Wirel. Commun. and Network. Conf. WCNC 1999*, New Orleans, LA, USA, 1999 pp. 1298–1302 (DOI: 10.1109/WCNC.1999.796947).

[8] T. Kunz and E. Cheng, "Multicasting in ad-hoc networks: Comparing MAODV and ODMRP", *Workshop on Ad Hoc Commun.*, Bonn, Germany, 2001 [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.5446&rep=rep1&type=pdf

[9] M. Qabajeh, A. Abdalla, O. Khalifa, and L. Qabajeh, "A survey on scalable multicasting in mobile ad hoc networks", *Wirel. Personal Commun.*, vol. 80, pp. 369–393, 2015 (DOI: 10.1007/s11277-014-2016-4).

[10] A. Rajeswari, "A mobile ad hoc network routing protocols: A comparative study", in *Recent Trends in Communication Networks*, P. Mitra, Ed. IntechOpen, 2020. pp. 1–24 (DOI: 10.5772/intechopen.92550).

[11] B. Yang, Z. Wu, Y. Shen, X. Jiang, and Sh. Shen, "On delay performance study for cooperative multicast MANETs", *Ad Hoc Networks*, vol. 102, pp. 1–14, 2020 (DOI: 10.1016/j.adhoc.2020.102117).

[12] S. Balaji, A. Rocha, and Y.-N. Chung (Eds.), *Intelligent Communication Technologies and Virtual Mobile Networks. ICICV 2019. Lecture Notes on Data Engineering and Communications Technologies*, vol. 33 pp. 169–178 and 607–617. Springer, Cham, 2020 (ISBN: 9783030283636).

[13] V. Vinya and G. Rao, "An energy efficient multicast route establishment using AODV with PSO algorithm and RSA for secured transmission", *Int. J. of Intell. Engin. and Syst.*, vol. 12, no. 5, pp. 257–266, 2019 (DOI: 10.22266/ijies2019.1031.26).

[14] A. Goyal and V. Sharma, "Design and implementation of modified local link repair multicast routing protocol for MANETs", *Int. J. of Scient. & Technol. Res.*, vol. 9, no. 2, pp. 2316–2321, 2020 [Online]. Available: http://www.ijstr.org/final-print/feb2020/Design-And-Implementation-Of-Modified-Local-Link-Repair-Multicast-Routing-Protocol-For-Manets.pdf

[15] L. Junhai, X. Liu, and Y. Danxia, "Research on multicast routing protocols for mobile ad-hoc networks", *Comp. Networks*, vol. 52, no. 5, pp. 988–997, 2008 (DOI:10.1016/j.comnet.2007.11.016).

[16] C. Morais, H. Gossain, and D. Agrawal, "Multicast over wireless mobile ad hoc networks: Present and future directions", *IEEE Network*, vol. 17, no. 1, pp. 52–59, 2003 (DOI: 10.1109/MNET.2003.1174178).

[17] Z. Xiaofeng and L. Jacob, "Multicast zone routing protocol in mobile ad hoc wireless networks", in *Proc. 28th Ann. IEEE Int. Conf. on Local Computer Networks LCN 2003*, Bonn/Konigswinter, Germany, 2003, pp. 50–159 (DOI: 10.1109/LCN.2003.1243122).

[18] Ch. Zhi, J. Cui, and L. Zhu, "Multicast routing algorithms based on Levy flying particle swarm optimization", *J. of Physics: Conf. Series 1453*, 2020 (DOI: 10.1088/1742-6596/1453/1/012005).

[19] A. Sufian, A. Banerjee, and P. Dutta, "A tree multicast routing based on fuzzy mathematics in mobile ad-hoc networks", in *Applications of Internet of Things*, J. Mandal, S. Mukhopadhyay, A. Roy, Eds. *Lecture Notes in Networks and Systems*, vol 137, pp. 107–117. Springer, 2020 (DOI: 10.1007/978-981-15-6198-6_10).

[20] K. Vanisrsee and V. Reddy, "Multicast cooperative routing for opportunistic data transfer in mobile ad hoc network", *J. of Electron. and Commun. Engin.*, vol. 12, no. 3, pp. 26–33, 2017 (DOI: 10.9790/2834-1203012633).

[21] L. Canourgues, J. Lephay, L. Soyer, and A. Beylot, "STAMP: Shared-tree ad hoc multicast protocol", *in Proc. of Military Commun. Conf. MILCOM 2006*, Washington, DC, USA, 2006 (DOI: 10.1109/MILCOM.2006.302031).

[22] D. Babu and M. Ussenaiah, "CS-MAODV: Cuckoo search and M-tree-based multiconstraint optimal multicast ad hoc on-demand distance vector routing protocol for MANETs", *Int. J. of Commun. Syst.*, vol. 33, no. 16, pp. 1–17, 2020 (DOI: 10.1002/dac.4411).

[23] T. Winter and P. Thubert, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", IETF ROLL working group, Feb. 2010 [Online]. Available: https://datatracker.ietf.org/doc/html/rfc6550 (retrieved 2021-9-30).

[24] C. Murthy and B. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall PTR, 2004 (ISBN: 9780133007060).

[25] J. Garcia and E. Madruga, “The core-assisted mesh protocol”, *IEEE J. on Selec. Areas in Commun.*, vol. 17, no. 8, pp. 1380–1394, 1999 (DOI: 10.1109/49.779921).

[26] I. Al-Mejibli, “Improve on-demand multicast routing protocol in mobile ad-hoc networks”, *J. University of Kerbala*, vol. 16, no. 1, pp. 351–363, 2018 [Online]. Available: http://iraqjournals.com/article_143909_b8dbdbc05b91967f48d085bf69d3a977.pdf

[27] J. Xie, R. Talpade, A. McAuley, and M. Liu, “AMRoute: Ad hoc multicast routing protocol”, *Mob. Netw. and Appl.*, vol. 7, no. 6, pp. 429–439, 2002 (DOI: 10.1023/A:1020748431138).

[28] J. Biswas, M. Barai, and S. Nandy, “Efficient hybrid multicast routing protocol for ad-hoc wireless networks”, in *Proc. 29th Ann. IEEE Int. Conf. on Local Comp. Networks*, Tampa, FL, USA, 2004 (DOI: 10.1109/LCN.2004.47).

[29] G. Walikar and R. Biradar, “Energy aware hybrid multicast routing in mobile ad hoc networks: zone-based approach”, *Int. J. of Mob. Netw. Design and Innov.*, vol. 8, no. 2, pp. 80–100, 2018 (DOI: 10.1504/IJMNDI.2018.092344).

[30] T. Shih, C. Shih, and C. Chen, “Location-based multicast routing protocol for mobile ad hoc networks”, *WSEAS Trans. on Computers*, vol. 7, no. 8, pp. 1270–1279, 2008 (DOI: 10.5555/1457999.1458015).

[31] X. Xiang, X. Wang, and Y. Yang, “Supporting efficient and scalable multicasting over mobile ad hoc networks”, *IEEE Trans. on Mob. Comput.*, vol. 10, no. 4, pp. 544–559, 2010 (DOI: 10.1109/TMC.2010.176).

[32] J. Jetcheva and D. Johnson, “Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks”, in *Proc. ACM Int. Symp. on Mob. AdHoc Network. & Comput. MobiHoc’01*, Long Beach, CA, USA, 2001 (DOI: 10.1145/501422.501423).

[33] M. Qabajeh, A. Hashim, O. Khalifa, and L. Qabajeh, “Geographical multicast quality of service routing protocol for mobile ad-hoc networks”, *J. of Engin. Lett.*, vol. 18, no. 3, pp. 212–225, 2010 [Online]. Available: http://www.engineeringletters.com/issues_v18/issue_3/EL_18_3_02.pdf

[34] A. Daniel, “Position based multicast routing protocol for ad-hoc wireless network using backpressure restoration”, in *Proc. 2nd Int. Conf. on Comp. Engin. and Technol.*, Chengdu, China, 2010 (DOI: 10.1109/ICCET.2010.5485544).

[35] H. Hussen, S. Choi, J. Park, and J. Kim, “Predictive geographic multicast routing protocol in flying ad hoc networks”, *Int. J. of Distrib. Sensor Netw.*, vol. 15, no. 7, pp. 1–20, 2019 (DOI: 10.1177/1550147719843879).

[36] R. Shankar and E. Ilavarasan, “Scalable multicasting through hexagonal zone based structure over mobile adhoc networks”, *J. of Internet Technol.*, vol. 19, no. 7, pp. 2111–2124, 2018 (DOI: 10.3966/160792642018121907014).

[37] “GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems”, UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/

[38] S. Corson and J. Macker, “Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations”, RFC 2501, 1999 [Online]. Available: http://www.ietf.org/rfc/rfc2501.txt

Liana Khamis Qabajeh received her B.Sc. in Computer Engineering from Palestine Polytechnic University (PPU), Palestine, in 2000 and joined the Engineering and Technology Faculty, PPU, as a research assistant. She received her M.Sc. in Computer Engineering from Jordan University of Science and Technology, Jordan, in 2005. Between 2005 and 2008, before pursuing her study, she was primarily involved in academic teaching and research at PPU. She has secured her Ph.D. in Computer Science in 2012 from University of Malaya, Malaysia. In 2012 she was appointed assistant professor at the Information Technology and Computer Engineering Faculty, PPU. Her current research interests include distributed systems and ad-hoc networks.

https://orcid.org/0000-0002-2801-0070
E-mail: liana_tamimi@ppu.edu
Faculty of Information Technology
and Computer Engineering
Palestine Polytechnic University
Hebron, Palestine