# WINDMILL: A Scalable Multicast Routing Protocol for Mobile Ad-Hoc Networks

Liana Khamis Qabajeh
Information Technology and Computer Engineering Faculty, Palestine Polytechnic University, Hebron, Palestine
Email: liana_tamimi@ppu.edu

*Abstract*—**Multicasting supports various applications that need a high collaboration and require data transmission to numerous destinations concurrently. In environments where nodes are continuously moving, as in mobile Ad-hoc networks, seeking efficient routes from a specific source leading to anticipated destinations became an important issue. In this paper a novel scalable multicast routing procedure for mobile Ad-hoc networks is proposed. Our new protocol, WINDMILL, tries to improve performance by introducing a hierarchal routing algorithm and dealing with the area as zones. Furthermore, WINDMILL tries to demonstrate better scalability, performance and robustness through applying the restricted directional flooding. WINDMILL utilizes the network partitioning to forward the route request and reply packets in an efficient way and avoid forwarding duplicate packets. A qualitative comparison between WINDMILL, multicast Ad-hoc on-demand distance vector (MAODV), on-demand multicast routing protocol (ODMRP) and location aware multicasting protocol (LAMP) protocols is presented in this paper. This comparison has considered the used routing category, main contribution, routing structure and network structure maintenance, request and reply packets sending mechanisms, route activation time, selected routes length, and data packet copies number. Our investigation reveals that WINDMILL will be able to achieve scalability by attaining reduced control overhead and low number of data packets copies even within large networks. Hence, WINDMILL can be a good choice for multicasting in Ad-hoc networks established for example among students on a campus or soldiers in a battlefield, where scalability is a key issue.**

*Index Terms*—**Ad-hoc networks, MAODV and ODMRP, mobile, multicast, performance evaluation, position-based, qualitative comparison, routing protocol, scalable, WINDMILL**

## I. Introduction

An Ad-hoc network differs from other wireless networks by its self-regulation multi-hop nature. Ad-hoc networks can be implemented in mountains, jungles and deserts [1]. Such networks require rapid deployment and dynamic reconfiguration. Upon using wireless communication, it is a crucial issue to reduce the power consumption and transmission overhead since nodes rely on batteries and wireless links usually has low-bandwidth [2], [3]. Effective routing is a key issue in Ad-hoc networks since nodes are moving rapidly in most cases [4]-[6].

Multicasting efficiently supports diverse applications that require close collaboration [7], [8]. Hence, it supports applications that involve simultaneous data transmission to a set of hosts identified by a single destination address. The design and details of a multicast routing protocol should take into consideration application requirements, design goals, and network properties [7], [9], [10]. Military battlefields, disaster recovery, rescue sites, emergency search, and distributed collaborative computing are some examples of multicast applications for the mobile Ad-hoc networks where members share information among themselves using their mobile devices [11]. Multicast group members may move resulting in a random and rapid topology change at unpredictable times [12]. Thus, tree reconfiguration schemes and membership information maintaining should be simple and keep low channel overhead [9], [13]-[15].

Constrained power, limited bandwidth, and mobile hosts make designing multicast protocols a challenging issue [16]. Position-based routing has been highly introduced in mobile Ad-hoc networks, due to the need for scalable and energy-efficient routing, in addition to the recent availability of small and inexpensive positioning instruments [14]. Numerous multicast routing protocols have been suggested for Ad-hoc networks. One of the most popular and benchmark protocols is multicast Ad-hoc on-demand distance vector (MAODV) [17]. Another protocol of interest is the On-Demand Multicast Routing Protocol (ODMRP) [14]. MAODV uses a multicast tree based on hard state information, while ODMRP constructs a mesh-based structure on soft state. Recently, Location Aware Multicasting Protocol (LAMP) [18] has been proposed. LAMP utilizes a hexagonal zone-based structure to maintain scalable multicasting.

In this paper, WINDMILL multicast routing model is proposed. WINDMILL suggests a hierarchal routing procedure to enhance performance of the routing protocol and distribute load by dealing with the area as zones. Furthermore, WINDMILL aims to assure better scalability and robustness via using the idea of Restricted Directional Flooding (RDF). Hence, zone leaders (ZLs) work also as position servers and each group member should keep ZL of its zone updated about its position.

Manuscript received January 13, 2021; revised March 12, 2021; accepted May 10, 2021.

Corresponding author: Liana Khamis Qabajeh (email: liana_tamimi@ppu.edu)


doi: 10.18178/ijeetc.10.5.341-354

WINDMILL consists mainly of five phases; structure setup, structure maintenance and membership update, route construction, route maintenance, and data forwarding. To reduce duplicate packets, our algorithm allows each ZL to deliver the request packet to at most two neighboring zones and to send only one reply packet.

A qualitative comparison between WINDMILL, MAODV, ODMRP and LAMP protocols is conducted in this paper. Many parameters have been taken into consideration including the used routing strategy, main idea and contribution, routing structure and network structure maintenance, Request and Reply packets sending mechanisms, multicast route activation time, length of the selected routes, and number of Data packet copies.

Through this research we are trying to answer the following research question; will the newly proposed multicast routing strategy help in achieving high level of performance and scalability compared to the other three protocols? Hence, we can set out our research hypotheses; that is, utilizing the proposed multicast routing strategy will improve performance and scalability.

It is expected that WINDMILL will be able to achieve scalability by preserving reduced packet routing load compared to other protocols. The cost of WINDMILL is a little bit longer paths passing through ZLs.

Hence, our main contributions in this paper can be summarized as follows:

- Presenting recent unicast and multicast routing protocols for Ad-hoc networks.
- Proposing a new scalable multicast routing protocol for Ad-hoc networks.
- Providing improved performance via introducing a hierarchal routing algorithm and dealing with the network as zones.
- Demonstrating better scalability, performance and robustness via applying restricted directional flooding.
- Conducting a qualitative comparison among the newly proposed protocol and other existing ones.

The rest of the paper is organized as follows. Section II introduces the existing and recent works on Ad-hoc unicast and multicast routing protocols, as well as describing the operation of MAODV, ODMRP and LAMP protocols. Section III presents our new protocol, WINDMILL. Section IV involves a qualitative comparison among these protocols. Section V discusses our findings. Finally, concluded remarks and future directions are discussed in Section VI.

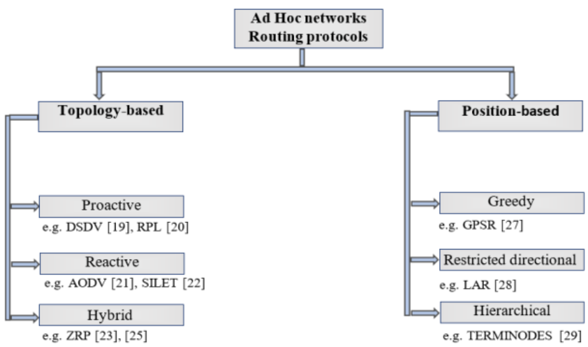


Fig. 1. Ad-hoc networks routing protocols categories.

## II. Background

In this section, mobile Ad-hoc networks unicast and multicast protocols are discussed in Subsections A. and B. respectively. Two of the most popular and benchmark protocols are MAODV and ODMRP. LAMP is a recently proposed scalable multicasting protocol. Hence, Subsections C through E present the details of these protocols.

### *A. Ad-Hoc Networks Routing Protocols Categories*

Ad-hoc networks routing protocols are generally classified into: topology-based and position-based (refer to Fig. 1). Topology-based category uses information about network links to accomplish forwarding packets. They are, in sequence, divided into: proactive, reactive, and hybrid protocols. In proactive protocols, such as destination sequenced distance vector protocol (DSDV) [19], nodes maintain network topology information via routing tables and periodically exchange routing information. Consequently, routes to every destination are available. This means that there is, roughly speaking, no delay prior to sending data. Another proactive protocol of interest is the routing protocol for low-power and lossy networks (RPL) [20]. However, proactive protocols seem less appropriate for Ad-hoc networks due to high control overhead as a consequence of periodic routing table updates. Reactive protocols conduct a route discovery procedure only when there is data to be sent. A route maintenance technique is needed for currently used routes only. Ad-hoc on-demand distance vector (AODV) [21] protocol is an example of reactive protocols. Another example is source-initiated link expiration time protocol (SILET) [22], which is a source-initiated reactive routing protocol that considers the predicted link expiration time while calculating the links weights. The destination selects the route having the minimum sum of the links weights. Proactive routing uses large network bandwidth, while reactive routing involves long route acquisition delay.

Hybrid routing, such as zone routing protocol (ZRP) [23], combines both approaches in a try to eliminate the delay associated with reactive routing and controlling the overhead associated with proactive routing [24]. Authors in [25] proposed a new hybrid routing protocol that utilizes advantages of both proactive and reactive approaches via allowing mobile nodes to flexibly run either a proactive or a reactive protocol considering their velocity and traffic.

On the other side, position-based protocols use information about nodes positions to help in forwarding packets. Hence, nodes should obtain their geographical position via Global Positioning System (GPS) for example. Also, the destination geographical position should be attained via a location service algorithm. Proposed position-based protocols are divided into: greedy forwarding, restricted directional flooding, and hierarchical routing protocols [26]. In greedy forwarding, such as greedy perimeter stateless routing (GPSR) [27], local topology is needed. Therefore, nodes periodically issue small beacons to enable their one-hop neighbors to maintain their positions [26]. Each node forwards a packet to its neighbor with the best progress towards the destination than itself. Greedy method is considered scalable since there is no need for route discovery and maintenance. Greedy forwarding has degraded performance in sparse networks, since forwarding node may not find a node towards the destination. Furthermore, proactive beaconing of one-hop neighbors has to be maintained at each node; which results in high congestion and consumes nodes energy. Moreover, greedy forwarding requires complex computation by nodes resulting in increased delay at intermediate nodes.

Restricted directional flooding, such as location aided routing (LAR) [28], tries to limit the flooding region. The sender broadcasts packet to all one-hop neighbors. Intermediate nodes, upon receiving a packet, compare their distance to the destination with the distance of their previous hop to the destination. In the case that the receiver node is closer to the destination, the packet is retransmitted. Otherwise, the packet is thrown down. In hierarchical protocols, two levels are used in a try to provide scalability. If the destination is close to the sender, packets are routed based on a proactive routing. Alternatively, greedy forwarding is utilized for long distance routing. Ad-hoc network properties such as constrained power and bandwidth, along with the need for scalable and energy-efficient protocols, justify utilizing position-based routing in such networks [26], [29], [30]. Terminal nodes framework (Terminodes) [29] is an example of hierarchical protocols.

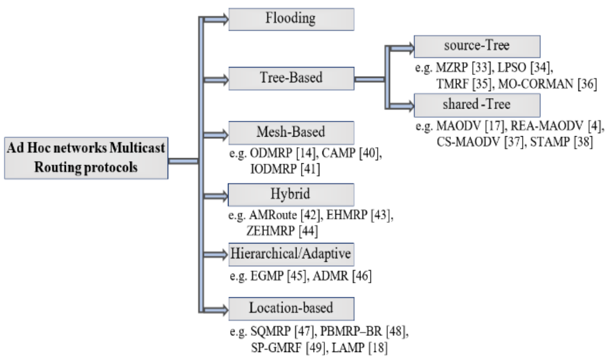


Fig. 2. Ad-hoc networks multicast routing protocols categories.

### *B. Ad-Hoc Networks Multicast Routing Protocols Categories*

Authors in [7] classified multicast routing protocols, considering the used delivery structure and connectivity maintenance strategy, into six categories: flooding, tree-based, mesh-based, hybrid, hierarchical/adaptive, and location-based. Fig. 2 summarizes these categories. One straightforward multicast technique is flooding. Flooding is the easiest way to perform multicasting since it does not need maintaining explicit multicast infrastructure. Upon initiating a multicast session, a source broadcasts the packet to the neighboring nodes. If a packet is not received before, a participating node simply broadcasts it to all its neighbors. If not, the packet is dropped. This procedure continues until the packet is sent throughout the entire network. Hence, data packets are rapidly spread throughout the network. Moreover, flooding achieves the minimum control overhead and is considered as the most reliable scheme. However, this comes at the expense of high data traffic. This is a result of duplicated packets in the network, which increases contention and wastes bandwidth. These problems become worse in large networks; i.e., flooding is not a scalable technique [7], [31]. Moreover, as packets are sent to all nodes, and interested nodes only accept the packets, this scheme is not used separately but it is utilized within other multicast protocols [32].

In tree-based multicast protocols, the multicast tree is created starting from the source of the data and links all the destinations. There is only one path from the source to each destination. Considering the number of trees per multicast group, a tree-based protocol is classified as a source-tree or shared-tree. In source-tree routing, the tree is rooted by the source node itself. Whereas in shared-tree routing, a sole tree that is rooted at a core node is shared by all sources in the group. In source-tree protocols, each multicast packet is forwarded via the most effective route from the source to each receiver. However, this method suffers from increased control overhead to keep numerous trees. At heavy loads, source-tree protocols perform better than shared-tree ones as they distribute traffic through different trees. Nevertheless, shared-tree protocols scale better and achieve reduced control overhead as they maintain only one group tree. Yet, shared-tree protocols suffer from the single point of failure; the core node. Compared to flooding, tree-based multicast mitigates bandwidth consumption. On the other hand, they suffer from lower robustness upon operating in highly mobile networks since there is only sole path between each source-member pair [7]. Multicast Zone Routing Protocol (MZRP) [33], multicast routing algorithms based on levy flying particle swarm optimization (LPSO) [34], tree multicast routing based on fuzzy mathematics (TMRF) [35] and multicast opportunistic cooperative routing in mobile Ad-hoc network (MO-CORMAN) [36] are examples of source-tree protocols. Whereas, MAODV routing protocol [17], reliable and energy-aware MAODV (REA-MAODV) [4], cuckoo search and M-tree-based MAODV (CS-MAODV) [37] and Shared-Tree Ad-hoc Multicast Protocol (STAMP) [38] are shared-tree protocols.

Contrasting to tree-based techniques, mesh-based approaches permit data packets to be sent to the same destination via different paths. Utilizing several routes between any source-destination pair offers improved protection against topological changes, which in turn improves successful delivery chance [31], [39]. Nevertheless, mesh-based protocols achieve lower multicast efficiency compared to tree-based protocols as a result of redundant routes [7].

Mesh-based approaches are used in frequently-changing topology networks. Upon a primary link breakage due to nodes mobility, no need to conduct a network structure reconfiguration since there are multiple paths. This eliminates frequent network reconfigurations, which in turn minimizes disturbing current sessions and decreases the control overhead. Route discovery is conducted using broadcasting [31]. ODMRP [14], Core Assistant Mesh Protocol (CAMP) [40] and improved ODMRP (IODMRP) [41] are instances for mesh-based multicast protocols. *Hybrid* protocols aim to attain improved performance and robustness through combining advantages of both tree-based and mesh-based techniques [31]. Similar to mesh-based protocols, numerous routes are chosen in a try to deliver data packets to their destinations. Tree-based approach is used in route setup to increase multicast efficiency. Some examples of hybrid protocols are Ad-hoc multicast routing protocol (AMRoute) [42], efficient hybrid multicast routing protocol (EHMRP) [43] and zone-based energy aware hybrid multicast routing scheme (ZEHMRP) [44]. However, as other routing protocols, they suffer from some problems. AMRoute, for example, suffers from unbalanced traffic and core vulnerability [32].

Hierarchical protocols aim to provide scalability and eliminate number of participating nodes via arranging nodes into a particular hierarchy. Recently, many multicast protocols have been proposed considering clustering. Examples of these protocols include efficient geographic multicast protocol (EGMP) [45]. Adaptive approaches adapt themselves to different environmental conditions. For example, adaptive demand-driven multicast routing protocol (ADMR) [46] adjusts itself in view of network mobility. Once the mobility becomes very high, ADMR switches to flooding to overcome links breakage. Location-based protocols assume the availability of location information of participating nodes. Packets' forwarding is performed considering the location of the direct neighbors and the projected destinations. So, the nodes having improved progress towards the destinations are selected, ensuring reduction of the number of participating nodes. Since location-based protocols scale well in large wireless networks, they have recently attracted the researcher's attention. In location-based unicast routing, the destination position is included in the packet header. This is not effective in multicast routing since multicasting targets group of members, and including positions of all multicast members in the forwarded packet results in a scalability problem [7]. Some examples of location-based protocols are scalable QoS multicast routing protocol (SQMRP) [47], position-based multicast routing protocol for Ad-hoc network using backpressure restoration (PBMRP-BR) [48], scalable and predictive geographic multicast routing scheme in flying ad-hoc networks (SP-GMRF) [49] and LAMP [18].

Despite that there are several multicast routing protocols, a reasonable solution for mobile Ad-hoc networks is still not evident and there still exist many issues that require further investigation. These problems include reliability, security, scalability and power consumption [7]. Detailed surveys of some recently suggested multicast routing protocols are presented in [7], [32], [50]. Based on these surveys, it has been observed that most of the existing protocols do not consider scalability issue. A crucial problem in such protocols is that control overhead may become huge if the network becomes dense, large or includes large number of destinations. Accordingly, in this research, the scalability and efficiency of multicast routing protocols have been considered.

Regarding the mobile Ad-hoc networks working group at the internet engineering task force (IETF), two popular and benchmark protocols have been proposed; MAODV and ODMRP. Both protocols are presented in the following subsections as the performance of most other protocols is compared to them [7].

### C. MAODV

The MAODV routing protocol [17] uses a broadcast route-discovery procedure to determine multicast routes when needed. Nodes in the network originate a route request (RREQ) packet if they wish to join a multicast group, or if they have data to forward to a multicast group that they do not have a route to. Only members of the anticipated group can reply to a join RREQ. When the RREQ is not a join request, any node having an up-to-date route to that group may respond. Upon receiving a join RREQ to a group that it is not participated in, or upon receiving a RREQ to a group and it does not know a route to, an intermediate node rebroadcasts this RREQ to its neighboring nodes [17].

Intermediate nodes, when receiving a RREQ packet, update their route table by recording the sequence number and the next hop for the source node. Nodes may later use this reverse route entry to transmit a response back to the source. Regarding join RREQs, another entry is added to the multicast route table. If the route is selected to be part of the multicast tree, this entry is activated. Nodes receiving a join RREQ for a multicast group may reply if they are members of the group tree and the stored sequence number is the same or larger than that contained in the RREQ. Upon deciding to respond, a node updates its route and multicast route tables by recording the next hop information of the requesting node. After that, it sends a request response (RREP) towards the source node. Nodes in the way to the source, upon receiving the RREP, create the forward path by adding both a route and a multicast route tables entries for the node that sent the RREP [17]. Fig. 3 shows the sent RREQ and RREP packets upon MAODV path discovery.

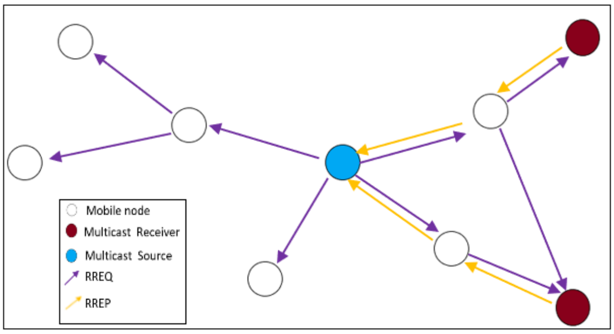


Fig. 3. Path Discovery in MAODV.

When broadcasting a RREQ for a specific group, a source node may receive many RREP packets. Hence, it waits a specific period and enables only the received route with the larger sequence number and smallest hop count to the nearest member of the tree. Accordingly, it enables the chosen next hop in its multicast route table, and sends an activation message (MACT) to that next hop. The next hop, in turn, enables the entry for the source in its multicast route table. If this node is a member, it stops propagating this message. However, if it is not a member of the multicast tree, it will have received one or more RREPs from its neighbors. So, it retains the best next hop for its route, unicasts MACT to that next hop, and activates the related entry in its multicast route table. This procedure continues till reaching the RREP originating node. Activation messages guarantee that there are no multiple paths to any tree node in the multicast tree. Hence, data is forwarded by nodes only along activated routes [17].

The group leader is the first member of the multicast group. This leader maintains the group sequence number and broadcasts it to group members via group hello packet. This packet contains the multicast group IP address and the sequence number of that group which is incremented every group hello. Nodes use this packet information to revise their request tables. As MAODV retains hard state in its routing table, it has to dynamically react to changes in this tree. If a member decided to leave the group, the multicast tree needs pruning. If a link failure is noticed, the node that is downstream of the break is in charge of repairing the failed link. If the tree cannot be reconnected, a new leader for the disconnected downstream node is selected [17].

### *D. ODMRP*

ODMRP [14] is a mesh-based protocol, rather than a conventional tree-based one. By utilizing a mesh, the drawbacks of multicast trees such as alternating connectivity, common tree reconfiguration, and non-optimal route are avoided [14].

In ODMRP, only part of the nodes forwards the multicast packets through scoped flooding. It utilizes on-demand procedures to maintain dynamic group membership and build routes. If a source has data to send and no routes to the group members are already established, source broadcasts a join-query packet to the whole network. This join-query packet is periodically broadcast to update membership information and routes [14] (refer to Fig. 4). To detect any potential duplicates, once intermediate nodes receive a join-query packet, they store source ID and sequence number within their message cache. Backward learning for the reverse path towards the source node is used; i.e., routing tables are updated with the node ID from which they received the message. The message is rebroadcast if it is not a duplicate and the Time-To-Live (TTL) is larger than zero [14].

Upon receiving a join-query packet, a multicast receiver sends a join-reply to its neighbors. Once a node gets a join-reply, it observes if the next hop of one of the entries matches its own ID. If yes, the node knows that it is part of the forwarding group, hence, it sets the forwarding group flag (FG_FLAG). Accordingly, it broadcasts its own join table created upon matched entries. The next hop node ID field is filled by getting information from nodes routing tables. So, each forward group member distributes the Join-Reply until reaching the source through the chosen shortest path [14].

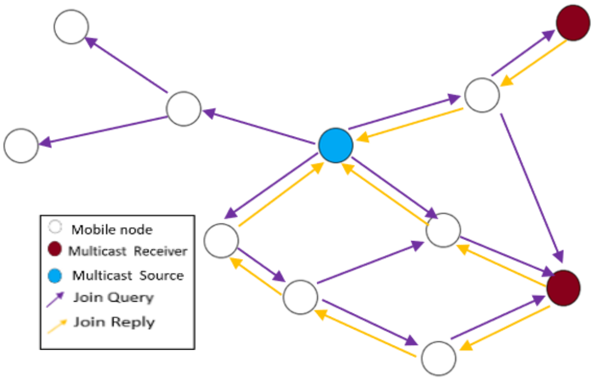

Fig. 4. Mesh creation in ODMRP.

Following to route construction process and forwarding group establishment (refer to Fig. 4), sources can send packets to destinations through these routes. As long as the source has data to be sent, it periodically sends join-query packets to refresh the routes. Upon receiving a data packet, a node resends it only if it is not a duplicate and the group FG_FLAG is not expired yet. This method reduces traffic overhead and avoids forwarding packets over expired routes [7], [14].

ODMRP adopts a soft state method to keep group members. Hence, no explicit control packets are sent to leave a multicast group. When a source decides to leave the group, it simply stops sending join-query packets. When a receiver no longer desires to receive from a certain group, it does not send the join-reply for that group. Nodes in the forwarding group are treated as non-forwarding nodes if no join tables received before they timeout. The reduction of channel/storage overhead and the relaxed connectivity make ODMRP more stable for mobile wireless networks [7], [14].

### *E. LAMP*

LAMP [18] aims to ensure scalability by reducing multicast routing overhead. LAMP involves three mechanisms; minimum-length tree construction, mobility-adaptive tree maintenance and zone-based greedy forwarding. LAMP creates a hexagonal zone-based structure to efficiently manage the group membership, and successfully track the multicast receivers' positions. The mobility-adaptive tree maintenance aims to optimize the performance via changing the tree structure according to continually varying topology. Thus, LAMP utilizes a mobility prediction approach to guess links lifetimes in the hexagonal zone.

At a certain node, multiple copies of a multicast packet have to be made to distribute it to different multicast receivers. Instead of considering the individual receivers positions information, LAMP enables the sender to choose a greedy zone for the receivers that are co-located in the same zone or neighboring zones. Hence, the zone-based greedy multicast forwarding scheme reduces the number of transmissions towards nearby-zones of multicast receivers and send a copy of the packet depending on the direction of the groups. Moreover, it chooses the multicast tree branches and selects the greedy forwarder for a group of receivers to reduce the overall path length. Hence, forwarding decisions are facilitated by zone members via providing their updated positions.

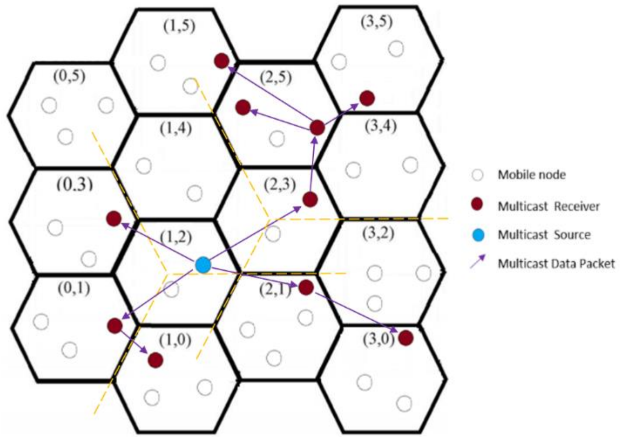


Fig. 5. LAMP greedy multicast routing.

LAMP forwards the packets to destinations along the established multicast tree among the nodes. At branching points, packet copies are disseminated to all the branches. When a source node S wants to send the data packets to a list of multicast receivers (refer to Fig. 5), S splits the network terrain into three regions with 120-degree angles. Then, it forwards a copy of the packet to each region that contains the set of multicast receivers, by identifying the ideal zone to reach the multicast receivers in each region. Then, S forwards the data packet towards the zone leader of the selected multicast greedy zone. This process is repeated in all the regions until it reaches all the multicast receivers.

Upon receiving a packet, a node retains the packet, if it is the destination or one of the next hops, otherwise, the packet is dropped. The neighboring nodes also conduct the same steps as S to choose the greedy zone. The greedy multicast routing aims to deliver the packets to several receivers via lower number of transmissions. When there is no such neighbor, LAMP escapes from the communication hole utilizing the perimeter forwarding mode.

Nodes in LAMP, perform somehow compound procedure to construct the hexagonal-zone structure and determine the specific zone that a node currently resides in. LAMP assumes that the nodes transmission range is the maximum distance between the nodes in a zone. This means that the area is divided into large number of small zones. Small zones mean that the number of zones leaders is large, resulting in increasing the overhead due to members and leaders' movement and failure, as well as new leader selection and transferring multicast table to the new leader.

The nodes in a zone select a leader by exchanging the beacon messages periodically, resulting in high overhead. Moreover, the used criteria to select the ZLs in LAMP is only selecting the node which is closer to the center of the zone. LAMP does not consider other important factors such as nodes processing, memory capacity and movement speed.

A member node in the routing zone maintains a location table that holds the geographical position information of the neighbors within its communication range. Periodically, each node broadcasts a beacon message to inform its location information to its one-hop neighbors. This periodic beaconing results in high process overhead and memory consumption of different nodes. Thus, all nodes in a zone know about all other locally nodes, not only the zone leader.

Source node initiates the session by sending the message to all the nodes in the network. Moreover, to re-initiate the session due to the source node mobility, source should terminate the current session by sending a message to the entire network. After its movement, the source should send a new message with its new location to re-initiate the session. Finally, greedy forwarding in not guaranteed to find a path especially in the existence of empty-zones.

## III. Proposed Protocol

In this section, WINDMILL routing model is represented. WINDMILL presents a hierarchal routing procedure to enhance performance and distribute load via dividing the area into zones. Furthermore, WINDMILL attempts to demonstrate better scalability and robustness towards continuous topological changes by means of restricted directional flooding. Accordingly, ZLs act also as position servers and each group member must keep ZL of its zone updated about its position.

Algorithm I shows the pseudocode for WINDMILL protocol. WINDMILL consists primarily of five phases; structure setup, structure maintenance and membership update, route construction, route maintenance, and data forwarding. These phases are presented in Subsection A. through Subsection E. Table I presents the used notations with WINDMILL, whereas Table II summarizes the used packet identifiers.

TABLE I: WINDMILL Notations

| Notation | Description |
|---|---|
| $IP_n$ | IP address of node *n* |
| $Pos_n$ | Position of node *n* |
| $Z_{[x,y]}$ | Zone number *x*, *y* |
| $Z_{[X,Y]}$ | Source node zone number *X*, *Y* |
| $D_{mov}$ | Movement distance allowed before sending PosUpdate |
| DD | Distance from a forwarding node to the destination |
| RDF | Restricted directional flooding |
| $ProbL_{n,xy}$ | Probability of electing node n as a ZL for its zone $Z_{[x,y]}$ |
| ZL | Zone leaders |
| $SN_n$ | Sequence number issued by node *n* |
| GID | Group number |
| $ZL_{[x,y]}$ | Zone leader of zone number *x*, *y* |
| $PosZL_{[x,y]}$ | Position of ZL of zone number *x*, *y* |
| $D_{TH}$ | Number of destinations inside a zone to decide to use RDF or ZBrd |
| ZBrd | Zone broadcast |
| $D_{cen}$ | ZL distance allowed from the zone center before sending ZLElect |

TABLE II: WINDMILL Packet Identifiers

| Packet identifier | Stand for |
|---|---|
| ZLProb | ZL Probability |
| ZLElect | ZL Election |
| JoinGroup | Join Group |
| PosUpdate | Position Update |
| SRREQ | Source Route Request |
| IRREQ | Internal Route Request |
| ERREQ | External Route Request |
| ZLPos | ZL Position |
| ZLQuery | ZL Query |
| LeaveGroup | Leave Group |
| RERR | Route Error |
| SRREP | Source Route Reply |
| IRREP | Internal Route Reply |

Algorithm I: Pseudocode for WINDMILL protocol

```
Start
Network structure setup phase: Dividing area into zones, electing ZLs and joining interested groups.
While not end of network lifetime
   {Structure maintenance and membership update phase: Consider nodes movement and changing their group membership.
    If any source has data to be sent
       {If there are no established routes between source and destinations
           {Route construction phase:
              Route discovery:
                 {Sending SRREQ from source node to its local ZL.
                  Sending ERREQ from source ZL to four neighbor ZLs.
                  Sending ERREQ from other ZLs to maximum two neighbor ZLs.
                  Sending IRREQ from ZLs only if there are interested nodes within current zone.
                 }
              Route setup:
                 Sending IRREP, ERREP and SRREP via reverse paths.
           }// If there are no established routes between source and destinations
       Else
           If there is any broken link in the established route
               Route maintenance phase: Sending RERR due to link breakage.
           Else
               Data forwarding phase: Source forwards data packets towards destinations through the established routes.
       }// If any source has data to be sent
   }// While not end of network lifetime
End
```

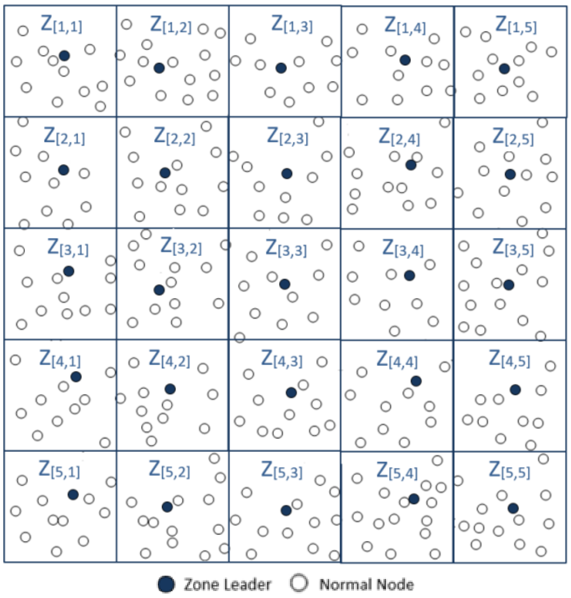


Fig. 6. Network structure after area division and ZLs election.

### A. Structure Setup

We assume cooperative nodes spread arbitrarily in a square-shape network and know their positions. During the structure setup phase nodes collaborate to split the network into zones and select an initial ZL for each zone. After that communication between ZL and interested nodes in joining a specific group is conducted. Subsections 1) and 2) discusses area division and ZLs election, and joining interested groups steps respectively.

*1) Area division and ZLs election*

At the beginning of the structure setup phase, the network is divided into several equal-size square-shape zones and initial ZLs for different zones are elected. Fig. 6 shows the network structure upon dividing the area into 5×5 zones and selecting ZLs. Each node knows the zone it belongs to using its position, the area coordinates, and number of zones. Node position is known via GPS, and the area coordinates and 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.

Upon electing ZLs, each node is assigned a weight representing its probability of being the ZL of its zone. The most important points upon selecting ZLs are the distance between the node and the center of its zone, movement speed and battery remaining life time.

Choosing a ZL that is close to the center of the zone makes the time needed for the communication between ZL and any node inside the zone almost the same. Choosing ZLs with low movement speed and high battery remaining life time increases the probability that the elected ZL will stay longer in the zone and decreases the probability of re-electing a new ZL within a short period of time. Another two important factors that are taken into consideration upon electing a ZL are the CPU processing power and the memory. ZLs with high CPU processing power and large memory significantly affect network performance since these ZLs are considered the operation bottleneck. Each node $n$ inside a specific zone $Z_{[x,y]}$ uses these factors to calculate the probability of itself to be elected as a ZL for this zone ($ProbL_{n,xy}$). Values of the weights of different parameters are chosen equally since we believe that they all are important upon selecting the ZL.

After calculating its probability to be elected as a ZL, each node sends a ZLProb message using zone broadcast (ZBrd). For instance, node $m$ residing in zone $Z_{[x,y]}$ sends the following message:

$IP_m$ <u>ZBrd</u> nodes in $Z_{[x, y]}$: [ZLProb, $IP_m$, $Z_{[x, y]}$, $ProbL_{m, xy}$]

ZLProb is the packet identifier, $IP_m$ is node $m$ IP address. Each node upon receiving the packet will process it only if it is inside the intended zone $Z_{[x, y]}$, otherwise the packet is dropped. The node with the highest probability in each zone will be the ZL of that zone. It sends ZLPos message to inform other nodes in its zone about its position. This message is also sent using ZBrd. Suppose that node n is the ZL of zone $Z_{[x, y]}$, then it will send:

$ZL_{[x, y]}$ <u>ZBrd</u> nodes in $Z_{[x, y]}$: [ZLPos, $Pos_n$, $IP_n$, $SN_n$, $Z_{[x, y]}$]

*2) Joining interested groups*

After that, only interested nodes send JoinGroup and PosUpdate messages to ZL of their zone to inform it that they are interested to join a specific group and tell it about their positions. These packets are sent using RDF. Utilizing RDF increases the probability of discovering a route compared to greedy. Moreover, this decreases overall overhead compared to blind broadcasting to the whole network. If a node $n$ at $Z_{[x, y]}$ is interested to join a session related to a multicast group number GID, then the following JoinGroup packet is sent:

$IP_n$ <u>RDF</u> $ZL_{[x, y]}$: [JoinGroup, GID, $IP_n$, $SN_n$, $Z_{[x, y]}$, DD]

The first two fields are the GID field that represents the multicast group ID, and the $IP_n$ field that states the IP of the interested node. Every node has Sequence number $SN_n$ that is increased monotonically with each JoinGroup packet. The fields ($IP_n$ and $SN_n$) are used to uniquely identify each JoinGroup packet. The field $Z_{[x, y]}$ represents the node zone number. The field DD is used to store the distance from the previous node to the destination. Hence, each node forwarding the packet, calculates the distance from itself to the local ZL and stores it in the DD field. The packet continues to be propagated restrictedly until reaching the intended ZL. When an intermediate node receives a JoinGroup packet with a ($IP_n$, $SN_n$) pair that has been processed earlier, the packet is not processed again. The interested node sends also a PosUpdate message using RDF, where the $Pos_n$ represents the Position of the interested node:

$IP_n$ <u>RDF</u> $ZL_{[x, y]}$: [PosUpdate, $Pos_n$, $IP_n$, $SN_n$, $Z_{[x, y]}$, DD]

### *B. Structure Maintenance and Membership Update*

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

Member nodes should also inform their ZL about their new position if they have moved a pre-defined distance ($D_{mov}$) from their last updated location. When a specific member is about to leave the boundaries of its zone, it should send a LeaveGroup to previous ZL. Then it sends a ZLQuery packet to ask about the ZL of the new zone. This packet is sent to 1-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 via sending JoinGroup and PosUpdate messages.

Regarding ZLs, a ZL sends ZLPos message to inform other nodes in its zone about its new position upon moving $D_{mov}$ from its last known position. This message is sent using ZBrd. If ZL decided to depart its zone, its distance from the zone center became higher than a pre-defined distance ($D_{cen}$), or its battery is about to turn off; it sends ZLElect packet to initiate a new ZL election. This packet is sent using ZBrd. Suppose that node n is the leaving ZL of zone $Z_{[x, y]}$, then it will send:

$ZL_{[x, y]}$ <u>ZBrd</u> nodes in $Z_{[x, y]}$: [ZLElect, $IP_n$, $SN_n$, $Z_{[x, y]}$]

Upon receiving this packet, every node inside the zone calculates its probability to become a ZL and a new ZL will be elected as discussed in the structure setup phase.

### *C. Route Construction*

This phase has different subphases. These subphases are presented in Subsections 1) through 3).

*1) Route discovery process*

When a source node has data to be sent to a multicast group, the subsequent steps take place:

(1) Upon deciding to initiate a multicast session, a source node directs a source route request (SRREQ) packet to its local ZL to enquiry about possible participating nodes. This packet is sent via RDF. If a source node s at $Z_{[X, Y]}$ is enquiring about nodes interested in joining a session related to a multicast group number GID, then the following SRREQ packet is sent:

$IP_s$ <u>RDF</u> $ZL_{[X, Y]}$: [SRREQ, GID, $IP_s$, $SN_s$, $Z_{[X, Y]}$, DD]

The first two fields are the GID field that represents the multicast group ID, and the $IP_s$ field that represents source node IP. Every node has $SN_s$ that is incremented with each request packet. $Z_{[X, Y]}$ field represents the source zone number. The field DD includes the distance from the previous node to the destination. Hence, the source node calculates the distance between itself and the local ZL and stores it in the DD field. Upon receiving SRREQ packet, an intermediate node computes the distance from itself to the required ZL and compares it with the DD field. If the intermediate node is further than the previous node, the packet is dropped. Otherwise, it stores its previous hop node to be used in the reverse path. Also, it modifies the DD field to represent the distance between itself and the destined ZL node. The intermediate node $i$ then adds its IP address ($IP_i$) to the packet to be used by its next hop node. For example, node $i$ will forward the following packet:

$IP_s$ <u>RDF</u> $ZL_{[X, Y]}$: [SRREQ, GID, $IP_s$, $SN_s$, $IP_i$, $Z_{[X, Y]}$, DD]

The SRREQ packet continues to be propagated restrictedly until reaching the intended ZL. When an intermediate node receives a SRREQ packet with a ($IP_s$, $SN_s$) pair that has been formerly processed, the packet is not processed again.

(2) When the ZL of the zone where the source is located ($ZL_{[X,\,Y]}$) gets the SRREQ packet, it sends an external route request (ERREQ) packet to the four neighbor ZLs. For example, the following ERREQ packet is sent from source zone $Z_{[X,\,Y]}$ to neighbor zone $Z_{[x,\,y]}$:

$$ZL_{[X,\,Y]}\ \underline{\text{RDF}}\ ZL_{[x,\,y]}: [\text{ERREQ}, \text{GID}, IP_s, SN_s, Z_{[X,\,Y]}, ZL_{[X,\,Y]}, ZL_{[x,\,y]}, PosZL_{[x,\,y]}, \text{DD}]$$

The GID, $IP_s$, $SN_s$ and $Z_{[X,\,Y]}$ fields are same as in SRREQ. The field $ZL_{[X,\,Y]}$ represents the IP address of the zone leader that is currently forwarding the packet, and the fields $ZL_{[x,\,y]}$, and $PosZL_{[x,\,y]}$ represent the IP address and position of the neighbor zone leader. The field DD stores the distance between the previous and the destination nodes. These fields are used as with SRREQ.

(3) 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. Each node will process the packet only if it is in the intended zone $Z_{[X,\,Y]}$, otherwise the packet is dropped.

$$ZL_{[X,\,Y]}\ \underline{\text{RDF/ZBrd}}\ \text{nodes in } Z_{[X,\,Y]}: [\text{IRREQ}, \text{GID}, IP_s, SN_s, IP_i, Z_{[X,\,Y]}, \text{ZBrd}]$$

The GID, $IP_s$, $SN_s$ and $Z_{[X,\,Y]}$ fields are same as in SRREQ. The field $IP_i$ represents the IP address of the node that is currently forwarding the packet (which is set initially as $ZL_{[X,\,Y]}$). This packet is sent using zone broadcast (ZBrd is set to 1) if the number of destination nodes in this zone is greater than $D_{TH}$. In zone broadcast, upon deciding to forward the packet, the node stores the IP address of its previous hop ($IP_i$) to be used in the reverse path. Also, it modifies the $IP_i$ field to be its own IP address and continues forwarding the packet. Alternatively, if the destinations number in this zone is fewer than or equal $D_{TH}$, RDF will be used (ZBrd is set to 0). In this case, $ZL_{[X,\,Y]}$ will prepare separate packet to each destination, and each node processing the packet will forward it only if it is closer to that destination.

(4) Upon receiving an ERREQ for the first time, the intended neighbor ZL continues the route discovery through searching for 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 spread until reaching all network zones using the forwarding strategy discussed in the following subsection.

*2) External route request packets propagation*

WINDMILL protocol takes benefit of the network partitioning to forward the ERREQ packets in order to discover the anticipated group members. The proposed forwarding strategy aims to reduce overhead and avoid forwarding duplicate packets. In this subsection, sending ERREQ packet among different zones is illustrated. It is the responsibility of the ZL nodes to decide whether to forward the ERREQ packet to neighbor zones or not.

Referring to Fig. 7, assume that the source node resides in zone $Z_{[4,\,2]}$. Firstly, the ERREQ packet is sent towards the edge of the four neighboring zones (zones $Z_{[3,\,2]}$, $Z_{[4,\,1]}$, $Z_{[5,\,2]}$ and $Z_{[4,\,3]}$ in our example). Many duplicate packets will be certainly produced, if each zone receiving the ERREQ packet forwarded it to its 4 neighbors. Our algorithm allows each ZL to deliver the packet to at most two neighboring zones (Refer to Fig. 7). The ZL considers the source zone number $Z_{[X,\,Y]}$, and the number of the intermediate zone $Z_{[x,\,y]}$ that is about to forward the packet.

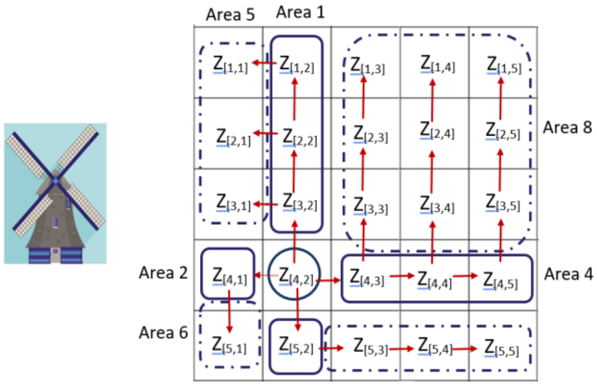


Fig. 7. Forwarding ERREQ packet in WINDMILL.

For illustration, assume that zone $Z_{[4,\,2]}$ is the zone that originally sent the packet. Here, the ZL node of zones $Z_{[4,3]}$, $Z_{[4,4]}$ and $Z_{[4,5]}$ (Area 4) forward the packet to the zones that are above and to the right (if any) of the current zone. In a following step, zones $Z_{[3,\,3]}$, $Z_{[3,\,4]}$ , $Z_{[3,\,5]}$, $Z_{[2,\,3]}$, $Z_{[2,\,4]}$, $Z_{[2,\,5]}$, $Z_{[1,\,3]}$, $Z_{[1,\,4]}$ and $Z_{[1,\,5]}$ (Area 8) send the packet only towards zones above them (if any). The same approach is utilized to forward packets to other network parts to reduce packets duplication. Fig. 8 shows the control packets exchanged during the route discovery phase of WINDMILL protocol.

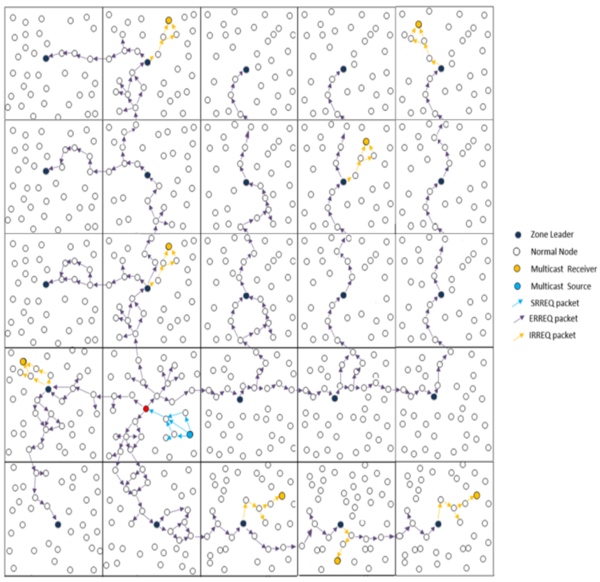


Fig. 8. Packets sent during the route discovery phase of WINDMILL.

*3) Route setup*

The next step, after propagating the request packets, is to setup the routes via sending the reply packets. The following steps are carried out during this phase:

(1) After forwarding the IRREQ packet and if it is interested in participating in the session, node *j* starts

setting up a route from the local ZL (ZL of the zone where node *j* currently resides) to itself by sending an internal route reply (IRREP) packet. Intermediate nodes forward this packet to the node from which it received the related IRREQ packet; i.e., the IRREQ with the same ($IP_s$, $SN_s$) tuple. This process continues till the packet reaches the intended ZL. For instance, node *j* sends the subsequent packet:

$IP_j$ Unicast $ZL_{[x, y]}$: [IRREP, $IP_s$, $SN_s$, $IP_j$, $IP_i$, $Z_{[x, y]}$]

This reply packet contains $IP_s$ and $SN_s$ to specify that this packet is a reply to the original unique IRREQ packet. It also holds IP address ($IP_j$) of the node interested in joining the session, IP address ($IP_i$) of the node that forwarded the corresponding IRREQ packet to it, and number of the zone $Z_{[x, y]}$ where the node currently resides.

(2) To reduce the overhead in the network, each zone leader $ZL_{[x, y]}$ sends only one external route reply (ERREP) to the neighbor ZL that forwarded the original ERREQ to it ($ZL_{[z, w]}$ for example). If the ZL itself is interested in the conducted session, it sends the ERREP packet immediately after forwarding the request packets (ERREQ and IRREQ). Otherwise, it sends the ERREP packet upon receiving the first ERREP or IRREP packet. This packet is sent via the reverse path towards the ZL node that sent the original ERREQ packet. The format of the ERREP packet from $ZL_{[x, y]}$ to $ZL_{[z, w]}$ is as follows:

$ZL_{[x, y]}$ Unicast $ZL_{[z, w]}$: [ERREP, $IP_s$, $SN_s$, $Z_{[x, y]}$, $Z_{[z, w]}$]

(3) To further reduce the overhead in the network, the source zone leader $ZL_{[X, Y]}$ sends only one source route reply (SRREP) to the source node *s*. If the ZL itself is interested in the conducted session, it sends the SRREP packet immediately after forwarding the request packets. Otherwise, it sends the SRREP packet upon receiving the first IRREP or ERREP packet. Each node sends SRREP packet to the previous hop from which it gets the original SRREQ packet, till the packet reaches node *s*.

$ZL_{[X, Y]}$ Unicast $IP_s$: [SRREP, $IP_s$, $SN_s$, $Z_{[X, Y]}$]

### D. Route Maintenance

Nodes' failure or movement during data forwarding may result in broken links causing some nodes not to receive several data packets. Our protocol considers the following points:

(1) Upon link breakage, the upstream node of the failed link issues a route error (RERR) packet to the upstream nodes telling them about the failure. Upon getting this packet, intermediate nodes delete the information about the downstream nodes, and resend the packet towards their upstream nodes. Moreover, the downstream nodes of the failed link delete the associated entry and free the resources after a specific time is passed without getting data from the upstream nodes.

(2) Upon receiving a RERR packet, a zone leader removes the corresponding entry and starts a new route discovery procedure towards the affected destination nodes as explained earlier (whether the affected destinations are local in the same zone or zone leaders of neighboring zones).

(3) Furthermore, upon receiving a RERR packet, the source concludes that the link between itself and the local zone leader does no longer exist. Consequently, the source node removes the corresponding entry in the routing table and starts a new route discovery procedure to rebuild the failed route towards the local zone leader.

### E. Data Forwarding

Source node waits for a specific time then sets up the routes to the interested nodes and starts sending data via these routes. Multicast data packets are forwarded along the established tree from the source to the ZL nodes. Upon receiving a data packet, ZL nodes send a copy of it to the members in their zone. Intermediate nodes merely pass data packets to their successors in the route.

## IV. Qualitative Comparison of MAODV, ODMRP, LAMP and WINDMILL

Table III summarizes the characteristics of the presented protocols. The four protocols are similar in some features. More specific, they are on-demand protocols so they determine routes only upon having data to be sent to the multicast group. In either protocol, route discovery utilizes request and reply phases and routing details are kept on intermediate nodes. Nevertheless, there are numerous substantial differences in the details and procedures of these protocols, which most properly result in significant performance differences.

TABLE III: STUDIED PROTOCOLS CHARACTERISTICS

| Performance parameter | MAODV | ODMRP | LAMP | WINDMILL |
|---|---|---|---|---|
| Type | Topology-based (on-demand) | Topology-based (on-demand) | Position-based (Zone-based greedy forwarding) | Position-based (Restricted Directional Flooding) |
| Main idea/ Contribution | Maintaining a hard state tree to achieve more efficiency compared to mesh, and avoid sending duplicate packets to receivers. | Maintaining a soft state mesh structure rooted from every source to provide diverse paths. | Achieving scalability and providing mobility adaptiveness to the multicast routing. | Solving scalability problem by reducing control overhead and data packets copies. |
| Routing structure and network structure maintenance | • Tree-based.<br>• Hard state; any link failure requires tree repair.<br>• Group leader sends periodic group hello messages. | • Mesh-based structure rooted from each source.<br>• Soft state.<br>• Offers alternative paths. | • Tree-based.<br>• Hard state; links breakages require tree repair or reconstruction.<br>• Increased overhead to handle nodes movement, failure, and new leaders' election; especially due to small zones. | • Tree-based.<br>• Hard state; links breakages require local tree repair.<br>• Does not require periodic group hello messages.<br>• Dividing area into zones and initial ZL election is conducted |

| | | | | |
|---|---|---|---|---|
| | | | • Increased overhead due to periodic messages needed to elect zones leaders and maintain location tables.<br>• Increased overhead due to sending the initiate and terminate sessions messages to all nodes in the network. | once.<br>• Updates such as joining and leaving groups, position update and new ZL election are conducted locally. |
| Request packets sending mechanism | Sent to the entire network. | Sent to the entire network. | • Initiate session messages are sent to the entire network. | • Sent using RDF or ZBrd.<br>• Sent only inside zones having interested destinations. |
| Reply packets sending mechanism | Unicasted through the only chosen route. | Broadcasted back to the source. | • When an interested node receives a new session message, it sends Join Request message to its zone leader to confirm the tree connection.<br>• When a new session message arrives, the zone leader records the number of the group and the source zone. | Unicasted through the only chosen route. |
| Multicast route activation | Receiver waits a specific time to receive multiple replies before issuing an activation message through the selected route. | Immediately after receiving request. | Source waits till receiving session initiation to start data delivery using greedy zone delivery. | Immediately after receiving request. |
| Selected routes length | Least hop count. | Shortest path. | A little bit longer since routes are forced to go through zones leaders, especially in the existence of empty zones. | A little bit longer since routes are forced to go through ZLs. |
| Data packet copies | Moderate. | Maximum, mesh structure causes large number of copies of data packets sent through different paths. | • Moderate.<br>• A copy of the data packet is made for each branch resulting in somehow higher number of copies especially due to large number of zones. | • Minimum.<br>• Only one copy is sent to zones with interested destinations, then each ZL forwards these packets to different destinations locally. |

Six main parameters have been considered, which are:

1) Routing structure and network structure maintenance: This parameter takes into account packets sent to construct and maintain network structure; including those sent to elect new leaders, update nodes positions and maintain group membership. Moreover, it considers the resulted control overhead from initiating and terminating sessions, constructing the routing structure and maintaining it in case of a link failure.
2) Request packets sending mechanism: Considers the resulted overhead from flooding the route request packets to the entire network or sending them using a specific technique.
3) Reply packets sending mechanism: Takes into consideration the caused overhead from broadcasting the route reply packets to the whole network or forwarding them via a specific procedure.
4) Multicast route activation time: Some protocols spend some time before activating multicast routes, whereas others activate routes instantly.
5) Length of the selected routes: The expected average number of hops of the paths discovered by a protocol.
6) Number of data packet copies: This parameter considers the needed copies of data packets through diverse paths to different destinations.

First of all, let us consider *routing structure and network structure maintenance*. ODMRP keeps a soft state mesh topology rooted from every source. Hence, ODMRP maintains alternative routes and there is no need to reconstruct the mesh in case of a link failure. Additionally, routes from source to different destinations are periodically revived by the source. So, based on the revive interval in ODMRP, the resulted route revive control overhead from different sources may raise scalability concerns. MAODV, on the other hand, utilizes a public bi-directional tree. In MAODV, the tree based on hard state and any link breakage triggers reactions to overhaul the tree. A group leader keeps up to date multicast tree information through issuing periodic group hello messages. Bi-directional trees are more efficient, compared to mesh, and avoid forwarding redundant packets to receivers.

LAMP constructs a hexagonal zone-based structure and utilizes zone-based greedy forwarding. LAMP results in increased overhead to handle members and zone leader movement, failure, and new leader selection; especially due to small zones and considering only nodes positions to select the ZLs. Moreover, periodic messages such as those used to electing zones leaders and maintains a location table causes increased overhead. Additionally, the needed messages to initiate and terminate sessions by a source node are sent to the entire network, resulting in higher overhead.

WINDMILL, like MAODV and LAMP, uses a hard state public bi-directional tree. Hence, links failures require local repair of the tree inside a specific zone or between two adjacent zones. On the contrary of MAODV, no need for sending periodic group hello messages. Regarding network structure maintenance in WINDMILL, dividing area into zones and initial ZL election is conducted once at the beginning of structure setup phase. After that any update such as joining and leaving groups, position update and new ZL election is conducted locally inside the intended zone and most properly using RDF. Hopefully, this will help in making control overhead under control.

Second, regarding *request packets sending mechanism*, both ODMRP and MAODV send the request packet to the entire network. Similarly, source node in LAMP sends initiate session message to the entire network. Whereas in WINDMILL, the request packets between zones are sent using RDF, and using RDF or ZBrd only inside zones having destinations inside them.

Third, concerning *reply packets sending mechanism*, ODMRP broadcasts the request reply back towards the source. Employing broadcasts, ODMRP permits multiple possible paths from the source back to the receiver. However, broadcasting reply packets requires non-interested intermediate nodes to drop the control packets and results in additional processing overhead. Instead, MAODV and WINDMILL choose only one route and unicast the reply. This strategy, reduces processing overhead on one hand but may result in losing route in case of intermediate node movement. It is expected that WINDMILL will send a smaller number of reply packets compared to MAODV, since only one route is chosen from ZL of a zone to ZL of its neighbor one, then routes are chosen locally from ZL to nodes inside that zone. While in MAODV separate routes are chosen to different destinations. In LAMP, when an interested node receives a new session message, it sends Join Request message to its zone leader to confirm the tree connection. When a new session message arrives, the zone leader records the number of the group and the source zone.

Fourth, let us discuss *multicast route activation time*. MAODV does not activate a multicast route instantly. A probable receiver waits a specific time letting numerous replies to be received. Then, receiver sends an activation message through the selected multicast route. On the contrary, ODMRP and WINDMILL activate routes immediately. Moreover, it is expected that WINDMILL will finalize the routing structure faster than ODMRP since number of reply packets received by each node is less which reduces time spent by these nodes in processing these replies. Source node in LAMP waits till receiving session initiation to start data delivery using greedy zone delivery.

Fifth, regarding *number of hops of the selected routes*, it is expected that routes in WINDMILL will be a little bit longer than both MAODV and ODMRP since routes are forced to go through ZLs. Moreover, routes in LAMP are a little bit longer than the other three protocols since routes are forced to go through leaders of small zones, and the situation becomes worse in the existence of empty zones.

Finally, concerning the *number of copies of the sent data*, we believe that WINDMILL will attain the minimum number. ODMRP mesh structure causes sending many copies of data packets through diverse paths. Regarding WINDMILL, it sends only one copy to zones with interested destinations, then each ZL forwards these packets to different destinations locally. LAMP forwards the data packets along the established multicast tree among the nodes using greedy zone delivery. A copy of the data packet is made for each branch resulting in somehow higher number of copies especially due to large number of zones.

To conclude, we expect that WINDMILL will be able to achieve low control overhead and reduced number of copies of data packets. This is achieved on the expense of slightly longer routes.

## V. Analysis and Discussion

The four discussed protocols are on-demand protocols. However, WINDMILL use a hard state shared bi-directional multicast tree passing through ZLs, hence, links breakages require local repair of the tree only inside a specific zone or between two adjacent zones. On the contrary of the other protocols, WINDMILL does not require sending periodic messages.

WINDMILL is expected to achieve the scalability issue by maintaining reduced control overhead contrasted to the other protocols. This is because nodes in MAODV and ODMRP are unaware of their and other nodes positions. Hence, all request packets are sent using broadcast to the entire network. Similarly, source node in LAMP sends initiate session message to the entire network. In WINDMILL, however, the request packets between zones are sent using RDF, and using RDF or ZBrd only inside zones having destinations inside them. Moreover, in WINDMILL, joining and leaving groups, position update and new ZL election are conducted locally inside the intended zone and most properly using RDF. This helps in making control overhead under control.

Compared to LAMP, WINDMILL divides the network area into small number of larger zones. The large zone area indicates that the number of ZLs is small, i.e., minimize the overhead due to member or leader movement or failure, and electing new leaders. Moreover, WINDMILL considers square-shape zones which make mapping each node to its current zone easier compared to hexagonal-zone. Additionally, LAMP suffers from increased processing and packet overhead and memory consumption of different nodes (not only the zone leader) to maintain information about all other local nodes, periodic messages needed to elect zones leaders, in addition to broadcasting initiate and terminate session packets. Finally, greedy forwarding in not guaranteed to find a path especially in the existence of empty-zones.

It is obvious that WINDMILL will be able to achieve low control overhead and reduced number of copies of data packets even inside large networks. This is owing to implementing restricted directional flooding rather than broadcasting, dealing with the area as zones and distributing load between several ZLs. This is achieved on the expense of a little bit longer routes.

This assures WINDMILL efficiency in achieving high level of performance and scalability compared to the other three protocols. Consequently, this proves our research hypotheses; utilizing the proposed multicast routing strategy will improve performance and scalability. Moreover, our results support the published works presented in the literature; i.e., position-based restricted directional flooding helps in achieving scalability.

## VI. Conclusions and Future Works

WINDMILL proposes a hierarchal routing model that seeks to enhance the performance and scalability via dealing with the network as zones. Our WINDMILL seeks to show improved scalability, performance and robustness by the use of restricted directional flooding. A qualitative evaluation of MAODV, ODMRP, LAMP and WINDMILL protocols have been elaborated in this paper. Our analysis shows that WINDMILL will be able to attain reduced control overhead and low number of copies of data packets even within large networks. This is achieved on the expense of a little bit longer routes. Thus, our research investigates that WINDMILL can be a suitable choice for achieving scalability and reducing overhead of multicast routing in Ad-hoc networks established for example among rescuers in a disaster area.

Owed to large number of nodes and large geographical area of Ad-hoc networks a simulation tool is to be used for evaluating and studying the performance of the proposed protocol and contrast it to existing protocols. Our next duties are to assess WINDMILL effectiveness in discovering routes considering different number of source nodes and multicast group members. Moreover, WINDMILL scalability will be tested within relatively high node mobility environments and large area networks.

One of the critical research limitations facing Ad-hoc networks is the real-environment implementation and testing particularly when having large number of nodes. Thus, we aim to implement and test our protocol via real implementation.

There are still many open research challenges facing Ad-hoc networks. More attention can be given to security issues, energy-efficiency and Quality-of-Service. Moreover, dynamic and adaptive routing protocols are exciting topics, in which, some routing protocol details can be changed considering the current state of the network.

## Conflict of Interest

The authors declare no conflict of interest.

## References


[1] F. Khan, A. Khan, S. Khan, I. Qasim, and A. Habib, "A secure core-assisted multicast routing protocol in mobile ad-hoc network," *Journal of Internet Technology*, vol. 21, no. 2, pp. 375-383, 2020.

[2] A. Prasanth and S. Pavalarajan, "Implementation of efficient intra- and inter-zone routing for extending network consistency in wireless sensor networks," *Journal of Circuits, Systems and Computers*, vol. 29, no. 8, pp. 1-25, 2020.

[3] A. Prasanth and S. Jayachitra, "A novel multi-objective optimization strategy for enhancing quality of service in IoT-enabled WSN applications," *Peer-to-Peer Networking and Applications*, vol. 13, pp. 1905-1920, June 2020.

[4] A. Tavizi and A. Ghaffari, "Tree-based reliable and energy-aware multicast routing protocol for mobile ad hoc networks," *The Journal of Supercomputing*, vol. 74, no. 11, pp. 6310-6332, 2018.

[5] M. Divya, S. Subasree, and N. Sakthivel, "Performance analysis of efficient energy routing protocols in MANET," *Procedia Computer Science*, vol. 57, pp. 890-897, December 2015.

[6] S. Shankar, H. Suresh, G. Varaprasad, and G. Jayanthi, "Designing energy routing protocol with power consumption optimization in MANET," *IEEE Tans. On Emerging Topics In Computing*, vol. 2, no. 2, pp. 192-197, 2014.

[7] M. Qabajeh., A. Abdalla, O. Khalifa, and L. Qabajeh, "A survey on scalable multicasting in mobile ad-hoc networks," *Wireless Personal Communications*, vol. 80, pp. 369-393, January 2015.

[8] A. Rajeswari, "A mobile ad hoc network routing protocols: A comparative study," *Recent Trends in Communication Networks*, pp. 1-24, July 2020.

[9] A. Aneiba and M. Melad, "Performance evaluation of AODV, DSR, OLSR, and GRP MANET routing protocols using OPNET," *International Journal of Future Computer and Communication*, vol. 5, no. 1, pp. 57-60, 2016.

[10] D. Ahmed and O. Khalifa, "An overview of MANETs: applications, characteristics, challenges and recent issues," *Int. Journal of Engineering and Advanced Technology*, vol. 6, no. 4, pp. 128-133, 2017.

[11] B. Yang, Z. Wu, Y. Shen, X. Jiang, and S. Shen, "On delay performance study for cooperative multicast MANETs," *Ad Hoc Networks*, vol. 102, pp.1-14, May 2020.

[12] V. Sarbhukan and R. Lata, "Impact of mobility and density on performance of MANET," in *Proc. Intelligent Communication Technologies and Virtual Mobile Networks Conference*, 2019, pp. 169-178.

[13] K. Thomas and E. Cheng, "Multicasting in ad-hoc networks: comparing MAODV and ODMRP," presented at the Workshop on Ad-Hoc Communications, Bonn, Germany, 2001.

[14] S. Lee, M. Gerla, and C. Chiang, "On-demand multicast routing protocol," in *Proc. IEEE Wireless Communications and Networking Conference*, 1999, pp. 1298-1302.

[15] V. Vinya and G. Rao, "An energy efficient multicast route establishment using AODV with PSO algorithm and RSA for secured transmission," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 5, pp. 257-266, 2019.

[16] A. Goyal and V. Sharma, "Design and implementation of modified local link repair multicast routing protocol for MANETs," *International Journal of Scientific & Technology Research*, vol. 9, no. 2, pp. 2316-2321, 2020.

[17] E. Royer and C. Perkins, "Multicast operation of the Ad-hoc on demand distance vector routing protocol," in *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 207-218.

[18] R. Shankar and E. Ilavarasan, "Scalable multicasting through hexagonal zone based structure over mobile Ad hoc networks," *Journal of Internet Technology*, vol. 19, no. 7, pp. 2111-2124, 2018.

[19] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234-244, 1994.

[20] A. Hassani, A. Sahel, and A. Badri, "FTC-OF: Forwarding traffic consciousness objective function for RPL routing protocol," *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 10, pp. 168-175, March 2021.

[21] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," presented at 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), New Orleans, Louisiana, 1999.

[22] N. Meghanathan, "A unicast MANET routing protocol to simultaneously minimize the stability-hop count tradeoff and end-to-end delay," in *Proc. 9th International Conference on Information Technology: New Generations*, Las Vegas, 2012, pp. 60-64.

[23] Z. Haas, M. Pearlman, and P. Samar, "The performance of query control schemes for the zone routing protocol," *ACM/IEEE Trans. on Networking*, vol. 9, no. 4, pp. 427-438, 2001.

[24] H. Toulni and B. Nsiri, "A hybrid routing protocol for VANET using ontology," *Procedia Computer Science*, vol. 73, pp. 94-101, October 2015.

[25] K. Soodl and N. Sah, "New unicast routing protocol using comparative study of proactive, reactive and hybrid protocols for MANET," *International Journal of Engineering Research and General Science*, vol. 2, no. 4, pp. 44-50, 2014.

[26] S. Kalhor, M. Anisi, and A. Haghighat, "A new position-based routing protocol for reducing the number of exchanged route request messages in mobile ad-hoc networks," presented at Second International Conf. on Systems and Networks Communication, Cap Esterel, French Riviera, France, 2007.

[27] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th ACM/IEEE Annual International Conference on Mobile Computing and Networking*, 2000, pp. 243-254.

[28] Y. Ko and N. Vaidya, "Location-aided routing (LAR) in mobile Ad-hoc networks," *ACM Wireless Network*, vol. 6, no. 4, pp. 307-321, 2000.

[29] L. Blazevic, L. Buttyan, S. Capkum, S. Giordano, J. Hubaux, and J. L. Boudec, "Self-organization in mobile ad-hoc networks: the approach of terminodes," *IEEE Communication Magazine*, vol. 39, no. 6, pp. 166-174, 2001.

[30] L. Qabajeh, M. Kiah, and M. Qabajeh, "A scalable and secure position-based routing protocols for Ad-hoc networks," *Malaysian Journal of Computer Science*, vol. 22, no. 2, pp. 99-120, 2009.

[31] L. Junhai, X. Liu, and Y. Danxia, "Research on multicast routing protocols for mobile ad-hoc networks," *Computer Networks*, vol. 52, no. 5, pp. 988-997, 2008.

[32] K. Farhan, F. A. Fattah, F. Al-tarawneh, and M. Lafi, "Survey paper on multicast routing in mobile Ad hoc networks," presented at IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, 2019.

[33] Z. Xiaofeng and L. Jacob, "Multicast zone routing protocol in mobile Ad-Hoc wireless networks," in *Proc. 28th Annual IEEE International Conference on Local Computer Networks*, 2003, pp. 150-159.

[34] C. Zhi, J. Cui, and L. Zhu, "Multicast routing algorithms based on levy flying particle swarm optimization," *Journal of Physics: Conference Series*, vol. 1453, no. 1, 2020.

[35] 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, and A. Roy, Ed. 2020, pp. 107-117.

[36] K. Vanisrsee and V. Reddy, "Multicast cooperative routing for opportunistic data transfer in mobile ad hoc network," *Journal of Electronics and Communication Engineering*, vol. 12, no. 3, pp. 26-33, 2017.

[37] 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," *International Journal of Communication Systems*, vol. 33, no. 16, pp. 1-17, 2020.

[38] L. Canourgues, J. Lephay, L. Soyer, and A. Beylot, "STAMP: shared-tree Ad-hoc multicast protocol," presented at Military Communications Conference (MILCOM), 2006.

[39] I. Khan, S. Madani, M. Saddique, and S. Malik, "Efficient overlay multicast routing for hybrid networks," *Malaysian Journal of Computer Science*, vol. 24, no. 2, pp. 99-120, 2011.

[40] J. Garcia and E. Madruga, "The core-assisted meshprotocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1380-1394, 1999.

[41] I. Al-Mejibli, "Improve on-demand multicast routing protocol in mobile Ad-hoc networks," *Journal University of Kerbala*, vol. 16, no. 1. pp. 351-363, 2018.

[42] J. Xie, R. Talpade, A. McAuley, and M. Liu, "AMRoute: Ad-hoc multicast routing protocol," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 429-439, 2002.

[43] J. Biswas, M. Barai, and S. Nandy, "Efficient hybrid multicast routing protocol for ad-hoc wireless networks," presented at 29th Annual IEEE International Conference, 2004.

[44] G. Walikar and R. Biradar, "Energy aware hybrid multicast routing in mobile ad hoc networks: zone-based approach," *International Journal of Mobile Network Design and Innovation*, vol. 8, no. 2, pp. 80-100, 2018.

[45] X. Xiang, X. Wang, and Y. Yang, "Supporting efficient and scalable multicasting over mobile Ad-Hoc networks," *IEEE Trans. on Mobile Computing*, vol. 10, no. 4, pp. 544-559, 2010.

[46] J. Jetcheva and D. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad-hoc networks," presented at ACM International Symposium on Mobile AdHoc Networking & Computing, Long Beach, Calfornia, USA, 2001.

[47] M. Qabajeh, A. Hashim, O. Khalifa, and L. Qabajeh, "Geographical multicast quality of service routing protocol for mobile ad-hoc networks," *Journal of Engineering Letters*, vol. 18, no. 3, pp. 212-225, 2010.

[48] A. Daniel, "Position based multicast routing protocol for AD-hoc wireless network using backpressure restoration," presented at the 2nd Int. Conf. on Computer Engineering and Technology, 2010.

[49] H. Hussen, S. Choi, J. Park, and J. Kim, "Predictive geographic multicast routing protocol in flying ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 15, no. 7, pp. 1-20, 2019.

[50] V. Sabaresan and S. Winster, "Multicast routing protocols for mobile ADHOC networks," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1, pp. 148-163, October 2019.



**Liana Khamis Qabajeh** received her B.Sc. from Palestine Polytechnic University (PPU), Palestine in 2000 in Computer Engineering and joined the Engineering and Technology Faculty, PPU, as a research assistant. She received her M.Sc. from Jordan University of Science and Technology, Jordan in 2005 in Computer Engineering. Between 2005 and 2008 before pursuing her study, she was primarily involved in academic teaching and research in PPU. She has secured her Ph.D. degree in Computer Science in 2012 from University of Malaya, Malaysia. In 2012 she has been appointed as assistant professor in Information Technology and Computer Engineering Faculty, PPU. She has been appointed as Master of Informatics program coordinator at PPU during 2016-2019. Her current research interests include distributed systems and Ad-hoc networks.