Wireless Netw (2010) 16:449–466
DOI 10.1007/s11276-008-0146-x

Hierarchical geographic multicast routing for wireless sensor networks

Dimitrios Koutsonikolas · Saumitra M. Das · Y. Charlie Hu · Ivan Stojmenovic

Abstract Wireless sensor networks comprise typically dense deployments of large networks of small wireless capable sensor devices. In such networks, *multicast* is a fundamental routing service for efficient data dissemination required for activities such as code updates, task assignment and targeted queries. In particular, *efficient* multicast for sensor networks is critical due to the limited energy availability in such networks. Multicast protocols that exploit location information available from GPS or localization algorithms are more efficient and robust than other stateful protocols as they avoid the difficulty of maintaining distributed state (multicast tree). Since localization is typically already required for sensing applications, this location information can simply be reused for optimizing multicast performance at no extra cost. Recently, two protocols were proposed to optimize two orthogonal aspects of location-based multicast protocols: GMR (Sanchez et al. GMR: Geographic multicast routing for wireless sensor networks. In Proceedings of the IEEE SECON, 2006) improves the forwarding efficiency by exploiting the wireless multicast advantage but it suffers from scalability issues when dealing with large sensor networks. On the other hand, HRPM (Das et al. Distributed hashing for scalable multicast in wireless ad hoc networks. IEEE TPDS 47(4):445–487, 2007) reduces the encoding overhead by constructing a hierarchy at virtually no maintenance cost via the use of geographic hashing but it is energy-inefficient due to inefficacies in forwarding data packets. In this paper, we present HGMR (hierarchical geographic multicast routing), a new location-based multicast protocol that seamlessly incorporates the key design concepts of GMR and HRPM and optimizes them for wireless sensor networks by providing both forwarding efficiency (energy efficiency) as well as scalability to large networks. Our simulation studies show that: (i) In an ideal environment, HGMR incurs a number of transmissions either very close to or lower than GMR, and, at the same time, an encoding overhead very close to HRPM, as the group size or the network size increases. (ii) In a realistic environment, HGMR, like HRPM, achieves a Packet Delivery Ratio (PDR) that is close to perfect and much higher than GMR. Further, HGMR has the lowest packet delivery latency among the three protocols, while incurring much fewer packet transmissions than HRPM. (iii) HGMR is equally efficient with both uniform and non-uniform group member distributions.

Keywords Wireless sensor networks · Geographic routing · Location-based multicast · Scalability · Energy

1 Introduction

Wireless sensor networks comprise deployments of large networks of small wireless-capable sensor devices. Popular applications of sensor networks are security monitoring, fire monitoring or wildlife observation. These applications typically require a *dense* deployment of sensors in order to

---

D. Koutsonikolas (✉) · S. M. Das · Y. C. Hu
School of ECE, Purdue University, West Lafayette,
IN 47907, USA
e-mail: dkoutson@purdue.edu

I. Stojmenovic
Electronic, Electrical & Computer Engineering, The University
of Birmingham, Birmingham, UK
e-mail: ivan@site.uottawa.ca

I. Stojmenovic
SITE, University of Ottawa, Ottawa, ON, Canada

minimize the possibility of holes in the deployment area and to maximize coverage. In such networks, *multicast* is a fundamental routing service for efficient data dissemination required for activities such as code updates, task assignment and targeted queries. In particular, *efficient* multicast for sensor networks is critical due to the limited energy availability in such networks.

Recently, several location-based multicast protocols for wireless networks have been proposed [1–3] which neither assume any unicast routing scheme nor build any distributed multicast routing structure. These protocols build multicast trees using location information available from global positioning systems such as GPS [4] and use geographic forwarding to forward packets down the multicast trees. Sharing the stateless nature of geographic forwarding, these protocols are *stateless*, as they carry encoded membership and location as well as tree information in each packet, so that the multicast membership and routing state do not have to be distributed as in traditional multicast protocols such as MAODV [5], ADMR [6] and ODMRP [7]. Hence stateless protocols are more robust and potentially more efficient than stateful protocols as they avoid the difficulty of maintaining distributed state in the presence of frequent topology changes due to node failure or mobility. Their efficiency over stateful protocols is more pronounced when the multicast group has a sparse membership, in which case the stateful protocols have to employ a high percentage of non-member forwarding nodes (i.e. large state) to maintain the tree or mesh.

Despite the high promise of the location-based protocols, there are two major challenges to further improvement of these class of protocols. First, these stateless protocols typically use geographic forwarding for data dissemination down each branch of the multicast tree, and hence they are not exploiting the multicast advantage of the wireless transmission. Second, as the size of the network increases, even for a sparse membership density for which these stateless protocols are good for, the overhead of the encoding membership in each data packet will become significant.

Most recently, two protocols were independently proposed to address the above two orthogonal challenges respectively: geographic multicast routing (GMR) [8] and hierarchical rendezvous point multicast (HRPM) [9]. GMR improves the forwarding efficiency of location-based multicast by exploiting the wireless multicast advantage (WMA). In GMR, each node along a multicast tree tries to send a data packet down multiple branches of the tree using a single broadcast transmission. With each transmission, a node specifies a set of its neighbors as relay nodes towards the destinations. The selection of relay nodes is done by optimizing the *cost over progress ratio*, which tries to maximize the progress made towards the destinations while minimizing the number of relay nodes. However, GMR includes information about all the selected relay nodes and all the destinations these relay nodes are responsible for in the data packet header, increasing the byte overhead associated with each data transmission, and thus limiting the scalability of the protocol. On the other hand, HRPM reduces the encoding overhead of location-based multicast by constructing a hierarchy of forwarding trees at virtually no maintenance cost via the use of geographic hashing, and by disseminating data down the forwarding trees in the hierarchy in a pipelined fashion. HRPM partitions a large multicast group into multiple manageable-sized subgroups, limiting the per-packet tree-encoding overhead to an application-specific constant, irrespective of the multicast group size. The partitioning is achieved by geographically dividing the deployment area into smaller cells, which form a hierarchy. While HRPM elegantly addresses the scalability problem of location-based multicast by reducing the encoding overhead, it can be inefficient in terms of packet transmissions, as it uses a simple data forwarding scheme where each node along the forwarding tree in each cell unicasts a data packet separately along each branch of the subtree rooted at it.

In this paper, we take the natural research path of exploring the combined benefits of the major design concepts of the above two protocols, and design a new protocol, called hierarchical geographic multicast routing (HGMR), which seamlessly integrates the orthogonal major design concepts of GMR and HRPM and optimizes them for wireless sensor networks by providing both forwarding efficiency (and hence energy efficiency) and scalability to large networks. In particular, HGMR starts with a hierarchical decomposition of a multicast group into subgroups of manageable size (i.e. encoding overhead) using HRPM's key concept *mobile geographic hashing*. Within each subgroup, HGMR uses GMR's *local multicast scheme* to forward a data packet along multiple branches of the multicast tree in one transmission. Thus, HGMR can simultaneously achieve energy-efficiency (through higher forwarding efficiency utilizing multicast advantage) and scalability (through low overhead hierarchical decomposition).

We first evaluate the performance of HGMR by comparing it to GMR and HRPM in an ideal environment, where there are no bit errors due to random noise and no packet collisions. Hence, *all* packets transmitted by a source are received by *all* destinations. Comparing different protocols in such an ideal environment isolates the impact of environments from the raw performance of different protocols and hence gives a clear picture of the forwarding efficiency and the encoding overhead of each protocol. Our evaluation results show that for group sizes up to 250 members, HGMR incurs only up to 8% more transmissions compared to GMR, when HRPM incurs up to 138% more transmissions, and HGMR maintains an encoding overhead less than 17%

(close to HRPM's 12%), when GMR's encoding overhead reaches up to 40%. Moreover, with a large network size of 1000 nodes, HGMR incurs fewer transmissions (13% less) even when compared to GMR, while maintaining a low encoding overhead of 15%, very close to that of HRPM (12%) and much lower than that of GMR (36%).

We then compare the performance of HGMR with those of GMR and HRPM in a realistic environment, where propagation loss is modeled using a realistic model, packets are corrupted due to noise, and packet collisions may occur when two or more nodes transmit simultaneously. In this case 802.11 MAC protocol is used to handle medium access. Comparing different protocols in such a realistic environment takes into account the effect of practical factors such as loss and contention and hence enables us to evaluate the performance of the protocols from the real applications' perspective, by looking at metrics such as Packet Delivery Ratio (PDR) and packet delivery latency. Our evaluation results show that HGMR outperforms the other two protocols in terms of both of these metrics. As the group size increases up to 250 members, HGMR maintains a PDR above 87%, same as HRPM (but with 120% fewer transmissions), and higher than GMR (74%), while maintaining the lowest packet delivery latency (13% lower than GMR and 20% lower than HRPM). As the network size increases to 1000 nodes, HGMR and HRPM maintain their performance, while GMR's performance is deteriorated resulting in a low PDR of only 60% and a high packet delivery latency (23% higher than HRPM's and 27% higher than HGMR's).

Finally, we evaluate the three protocols in two scenarios with non-uniform group member distribution with large imbalance in the number of group members among different cells. We show that HGMR maintains its superiority over the other two protocols. It achieves a PDR very close to HRPM, and much higher than GMR, while it incurs the lowest number of transmissions among the three protocols, which also results in the lowest delivery latency.

The design of HGMR effectively provides a general framework for simultaneously handling scalability and different aspects of energy efficiency in location-based multicast for wireless sensor networks. Scalability is always achieved by the use of a hierarchy via geographic hashing. In this paper we used GMR for data forwarding within each cell of the hierarchy in order to reduce the total number of transmissions. Different protocols can be used in place of GMR, based on the specific metric we want to optimize. For example, balanced energy consumption can be achieved by replacing GMR with a protocol that rotates between alternative forwarding trees within each cell in the hierarchy. Also, sleep-wake MAC protocols for sensor networks (e.g., [10–13]) that periodically put some nodes within each cell to sleep can be used to conserve energy.

The rest of the paper is organized as follows. Section 2 formulates the location-based multicast problem. Section 3 presents the HGMR protocol. Sections 4, 5 and 6 evaluate HGMR in ideal and realistic environments. Section 7 discusses related work and Sect. 8 concludes the paper.

2 Design issues

The multicast problem deals with transmission of information from a node to all members of a group while optimizing a certain application specific metric such as bandwidth cost or delay. In a wireless sensor network equipped with a positioning system such as GPS [4], each node can determine its own geographic location. Location information can be exploited to provide location-based multicast. Location-based multicast protocols encode the membership as well as tree information in each packet so that membership/forwarding state is not distributed as in traditional multicast protocols such as ADMR [6] or ODMRP [14]. In the following, we discuss the three components of a location-based multicast protocol for multihop wireless networks.

2.1 Group membership and location management

An efficient scheme for the management of group membership and locations is critical to the efficiency and scalability of location-based multicast, since nodes can potentially move or fail in mobile ad hoc networks or sensor networks. To manage the group membership, group members can either multicast their membership/locations to all other group members [2], or send their updates to the multicast root which can disseminate the updated group member location along with data packets down the multicast tree. Moreover, either only the location of the group members [2] or of all the nodes in the network [1] are required depending on the nature of the multicast tree used.

2.2 Multicast tree construction

Once the group membership and location information are obtained, the multicast tree can be constructed in at least three ways. (1) The multicast root node locally constructs a (complete) physical tree [1] consisting of group member nodes and other nodes en-route between the member nodes. This requires the multicast root to know the location of all nodes in the network. (2) The multicast root node locally constructs a (complete) overlay tree [2] consisting of only group member nodes, and each overlay hop is routed using some underlying unicast routing protocol such as geographic routing. (3) The multicast root node locally calculates only the first physical hop(s) of data dissemination tree based on local knowledge of its physical neighbors [8]. This approach can easily exploit the Wireless Multicast Advantage (WMA) by sending a data packet down multiple branches of the multicast tree using one broadcast transmission.

Many graph algorithms exist for the construction of multicast trees. These tree construction algorithms exploit the correlation between geometric distance and network distance (number of routing hops), i.e., longer geometric distance implies more network hops [2], and typically use geographic distances between nodes as edge weights.

2.3 Data delivery

The data delivery mechanism closely depends on the tree construction scheme discussed above. For example, in case (2), the vast majority of location-based protocols assume a greedy geographic forwarding algorithm as the routing protocol, where each node periodically announces its IP address and location to its one-hop (within the radio transmission range) neighbors, and each node maintains the IP and location information of its neighbors. Each packet being routed contains the destination address in the IP header and the destination's location (x- and y-coordinates) in an IP option header. To forward a packet, a node consults its neighbor table and forwards the packet to its neighbor closest in geographic distance to the destination's location. The above greedy geographic forwarding can lead to a packet reaching a node that does not know any other node closer to the destination than itself. This indicates a hole in the geographical distribution of nodes. Recovering from holes can be achieved using *face-routing* [15–17].

3 Hierarchical geographic multicast routing

In this section, we present the design of HGMR. HGMR seamlessly incorporates the key design concepts of localized multicast of GMR [8] and hierarchical membership management using mobile geographic hashing of HRPM [9], and provides a hierarchical multicast protocol that exhibits the strength of both protocols: efficiency and scalability.

3.1 Overview of GMR and HRPM

3.1.1 GMR

Geographic multicast routing (GMR) [8] exploits the wireless multicast advantage to improve the forwarding efficiency of previous location-based (and hence stateless) multicast protocols. Similarly as in previous protocols, it assumes centralized membership management at the multicast root. Differently from in previous protocols, each node along the multicast tree tries to send a data packet down multiple branches of the multicast tree using one broadcast transmission.

In more detail, each forwarding node propagating multicast data needs to select a subset of its neighbors as relay nodes towards destinations. With GMR, nodes make this selection using a greedy heuristic which optimizes the *cost over progress ratio*. GMR decouples the cost of the multicast tree in terms of transmissions (which it tries to minimize using broadcast) and the progress made towards the destinations in terms of geographic distances. This decoupling makes it easy to model multicast tree construction as an optimization problem (with respect to the progress made towards the destinations). However, the use of the ratio as the final metric achieves a good tradeoff between the two individual metrics.

The *cost* in GMR is equal to the number of selected neighbors. Such a cost function tries to minimize the bandwidth consumption down the multicast tree, which is proportional to the number of forwarding nodes. On the other hand, the *progress* is calculated based on the idea of geographic forwarding, as the overall reduction of the remaining distances to the destinations. Such a neighbor selection scheme achieves a good tradeoff between the cost of the multicast tree and the effectiveness of data distribution.

We explain the cost over progress ratio metric with the help of Fig. 1. In this figure, assume that node C, after receiving a multicast message, is responsible for destinations D_1, D_2, D_3, D_4, D_5, and it is considering its neighbors A_1 and A_2 as possible relay nodes. Note that in the general case, D_1, D_2, D_3, D_4, D_5 are more one hop away from A_1 or A_2. The current total distance for multicasting is

$$T_1 = |\overline{CD_1}| + |\overline{CD_2}| + |\overline{CD_3}| + |\overline{CD_4}| + |\overline{CD_5}| \tag{1}$$

If C selects A_1 as the relay node responsible for D_1, D_2, D_3 and A_2 as relay node responsible for D_4 and D_5, the new total distance is

$$T_2 = |\overline{A_1D_1}| + |\overline{A_1D_2}| + |\overline{A_1D_3}| + |\overline{A_2D_4}| + |\overline{A_2D_5}| \tag{2}$$

and the progress made is

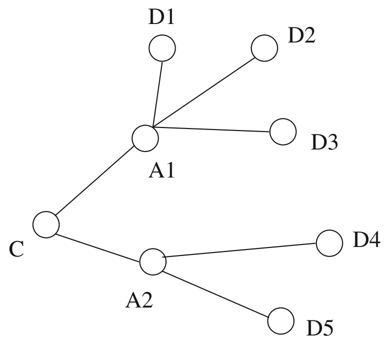


Fig. 1 Forwarding node selection in GMR

$$\Delta T = T_1 - T_2 \tag{3}$$

Hence, the cost over progress ratio for the forwarding set $\{A_1, A_2\}$ is

$$R = \frac{2}{T_1 - T_2} \tag{4}$$

Note that the above scheme is completely localized. Each node C only needs to know the locations of the destinations for which it is responsible and the locations of its one-hop neighbors; it does not require information about the topology of the whole network.

In some cases, a node C currently routing a data message may not have any neighbor providing advance towards some of the destinations. This situation is known as a local optimum for the greedy mode described above and it is handled using a technique similar to face routing used by geographic unicast protocols [15]. The destinations for which no progress can be made are put in a *multicast face-list* and they are routed in *perimeter mode* (in contrast to the normal *greedy mode* of operation) as follows. For each of those destinations, node C applies independently face routing [15] and decides the next hop for that destination. Face routing continues for each destination until a node can make progress for that destination; that node then removes the destination from the multicast face-list and switches back to greedy routing.

In the protocol implementation, GMR adds a header to data messages to allow neighbors to realize they are selected as relay nodes. Each node that forwards a message includes in the header the IDs of the neighbors it has selected as relay nodes and the IDs of the destinations each of the selected relay nodes is responsible for in greedy or in perimeter mode. The message is then *broadcast* and it can be received by all of its neighbors due to the broadcast nature of the wireless channel. This reduces the total number of transmissions and hence the total energy consumed, since each node along the multicast tree only needs to send a single message in order to deliver a multicast data packet to multiple relay nodes.

3.1.2 HRPM

Hierarchical rendezvous point multicast (HRPM) [9] reduces the encoding overhead by employing two key design concepts: (1) Use of hierarchical decomposition of multicast groups and (2) Leveraging geographic hashing to construct and maintain such a hierarchy efficiently. Figure 2 shows the major components of HRPM.

The main design goal of HRPM is to limit the per-packet overhead to an application-specified constant (ω), irrespective of the group size G. It achieves this by recursively partitioning a large multicast group into manageable-sized subgroups in which the tree-encoding overhead satisfies the ω constraint. This partitioning is achieved by geographically dividing the deployment area into smaller and smaller cells, which form a hierarchy with the root representing the entire region. Every cell in the hierarchy has an AP (Access Point) and the entire region has an RP (Rendezvous Point). All members in a leaf cell of the hierarchy form a subgroup and are managed by that cell's AP. Groups of APs are managed recursively, i.e., by the APs of their parent cells. Finally, APs belonging to the highest level of the hierarchy are managed by the RP. Das et al. [9] showed that a 2-level hierarchy is enough even for very large-size groups, up to 5800 members. In this case, the area is divided in d^2 cells, each with one AP, and the d^2 APs are managed by the RP. The parameter d is called *decomposition index*.

To avoid the need of keeping track of the AP/RP nodes, needed in both membership management and data dissemination, which would require an external location service, HRPM adopts the idea of geographic hashing to reduce the maintenance of AP/RP nodes at virtually no maintenance cost. The role of each AP as well as the RP is mapped to a unique geographic location, via some simple hash function, the node that is currently closest to that location then serves the role of AP/RP, and routing to the AP/RP is conveniently achieved via geographic forwarding. There are rare cases in which messages sent to the RP/AP from different nodes may not converge to a single node. To solve this problem, when a node A receives the first packet from another node, and it thinks it is the RP/AP, it buffers the packet and starts an expanding ring broadcast search for any other node in the neighborhood, which also thinks it is the RP/AP. If such a node is found, A relays the buffered packet to that node.

When a source has data packets to send, it first hashes the multicast group's identifier to obtain the location of the RP. It then contacts the RP and obtains the group membership vector, which specifies which cells have members (or in other words which of the APs are active). After that, the source builds an overlay tree, the *Source* → *APs* tree, considering each active AP as a vertex in a topology graph, and it sends data packets along the branches of this tree, using geographic forwarding. Each AP also builds an overlay tree, the *AP* → *Members* tree, whose vertices are the members in that AP's cell and forwards the data along this tree, also using geographic forwarding. As [9] showed, a minimum spanning tree (MST) achieves the best tradeoff among bandwidth efficiency, computational complexity, and location management overhead. Both the source and the APs use *unicast* to forward data packets, which means the same packet is sent over each branch of a subtree, with a different header, specifying the path it should follow along that subtree.

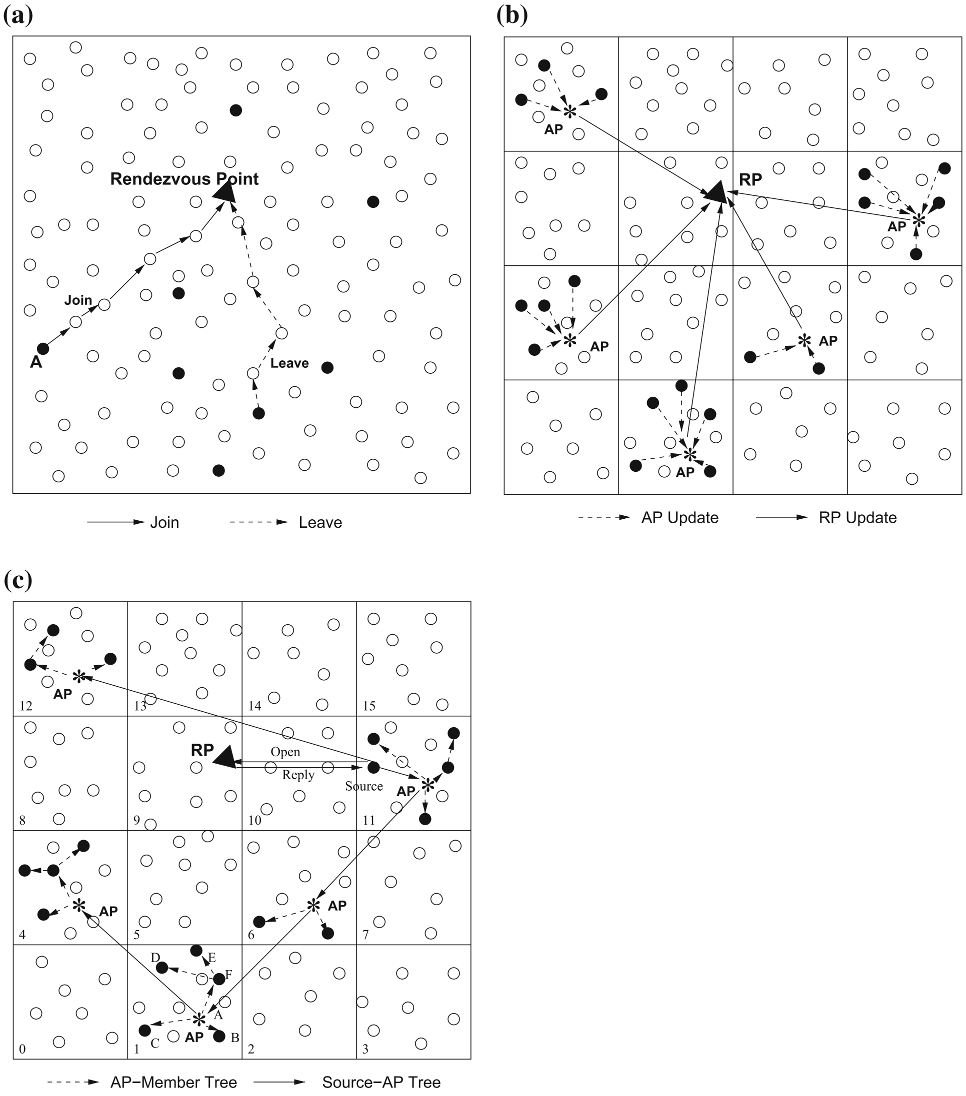


Fig. 2 Group management, tree construction, and data delivery in HRPM. (**a**) Rendezvous point group management in HRPM ($d = 1$). (**b**) Location updates in HRPM ($d = 4$). (**c**) Data delivery in HRPM ($d = 4$)

Similarly to previously proposed geographic routing protocols, HRPM has to deal with *holes* in the network topology. Holes can occur in two different cases in HRPM, when routing to a node and when routing to a hashed location. The first case can happen when the *AP* → *Members* tree is traversed to deliver data to individual group members. This case is similar to geographic unicast routing, and is handled using *face-routing* [15–17]. The second case can happen when the *Source* → *APs* tree is traversed to deliver data to APs. This case is slightly more complicated, since the node that encounters the hole has to distinguish whether the hole is en route to the hashed location or the hashed location is inside the hole. Similar to the first case, the node starts face routing. If the packet traverse around the face and comes back to the node, then that node becomes the AP.

We now briefly present the procedure for calculating the decomposition index d for the typical case of a two-level hierarchy. Assume that the total number of group members is G and the cost of encoding the node ID and its location is C bytes. Since the deployment area is divided into d^2 cells, the *Source* → *APs* tree has at most d^2 members, and the per-packet overhead is $d^2/8/f$ bytes, where f is the average fan-out of the overlay tree at the root. Each *AP* → *Members* tree has on average G/d^2 members, and thus, the per-packet encoding overhead is at most $C \cdot \frac{G}{d^2}/f$ bytes. d is calculated based on two constraints. The first constraint requires that the worst case encoding overhead in the *AP* → *Members* tree be less than ω bytes. With a worst-case fan-out from the tree root equal to 1, this constraint becomes:

$$C \cdot \frac{G}{d^2} \leq \omega \qquad (5)$$

The second constraint requires that the worst case encoding overhead in the *Source* → *APs* tree also be less than ω bytes, i.e.,

$$\frac{d^2}{8} \leq \omega \tag{6}$$

The *RP* first evaluates (5) to select a value for d that is large enough to satisfy that constraint. It then checks if this value satisfies (6). For example, with a multicast group of size 100, using (5), with $\omega = 102.4$ bytes (20% of a 512-byte packet), and $C = 12$ bytes, we obtain $d \geq 3.42$. As the value $d = 4$ satisfies (6), we divide the network into 16 cells. Detailed simulations in [9] showed that the values of d chosen based on the above analysis yield optimal performance.

When the group size changes, the RP may decide to adjust d in order to satisfy constraint (5). In that case, it multicasts a *NOTIFY* message with the new value for d to all members, using the current hierarchy. Upon receiving such a message, each member generates the hashed location for its new AP, and it sends an update to it. The new APs then send the aggregated membership to the RP.

3.2 Hierarchical geographic multicast routing (HGMR)

The overview above shows the strength and weakness of GMR and HRPM protocols. On one hand, GMR reduces the number of transmissions required to send a multicast data packet from the source to all destinations, since at each hop the packet is broadcast to all neighbors. However, this means that information about all the destinations and all the selected relay neighbors has to be included in the packet header. Assuming a reasonable node density, if the number of multicast members grows too large, the byte overhead associated with each packet may increase to unacceptable levels. On the other hand, HRPM efficiently reduces the byte overhead associated with each data packet by dividing a large group into multiple subgroups. However, HRPM is inefficient in terms of packet transmissions, since at each node along the *Source* → *APs* or the *AP* → *Members* tree, the same data packet is unicast to possibly more than one subtree. Unicasting the same data packet more than once not only consumes bandwidth, which is limited in wireless networks, but also exhausts faster the nodes energy which is limited, especially in sensor networks.

In this section, we propose Hierarchical Geographic Multicast Protocol (HGMR) which seamlessly combines the scalability (low encoding overhead) of HRPM with the forwarding efficiency of GMR. The integration of the unique features of the two protocols poses a few interesting challenges. The solution to reducing the encoding overhead is constructing a hierarchy of subgroups, similar to HRPM. For the delivery of the data to each of the subgroups, however, we could use either HRPM's (unicast-based) or GMR's (broadcast-based) forwarding strategy. GMR's strategy has the highest gain when the multicast member density is large; in that case, the benefit from broadcasting a packet instead of unicasting it to each member is maximized. For the *Source* → *APs* overlay tree, the AP density is expected to be low (one AP per cell), hence the benefit from using broadcast-based forwarding is not expected to be large. In addition, for large networks, some *APs* will be far from the source and the overlay paths to them will include many hops, reducing reliability of message delivery (since the wireless medium is inherently unreliable). Hence, using unicast-based forwarding for such long paths has one more advantage, since unicast MAC protocols usually incorporate a hop-by-hop reliability mechanism (e.g., ACKs and retransmissions in 802.11), in contrast to broadcast. On the other hand, within each cell, the density of multicast members is expected to be large, while the number of hops to each of them small. In this case, broadcast-based forwarding is preferred, since it can offer a significant reduction in the number of transmissions.

Based on the above observations, we now proceed to describe the new protocol. HGMR divides the multicast group into subgroups using the mobile geographic hashing idea proposed in HRPM: the deployment area is again divided into a number of cells; in each cell there is an AP responsible for all members in that cell, and all APs are managed by an RP. Membership management in HGMR is very simple and of almost zero cost, thanks to the static version of the nodes and the use of geographic hashing. To join a hierarchically decomposed multicast group, a node generates the hashed location for the RP and sends a *JOIN* message to that location. After receiving the value of decomposition index d from the RP, the node invokes the hash function with d and its location, to obtain the hashed location of the AP of the cell it belongs to. It then sends an *UPDATE* message to the AP. This completes the join process. To leave a group, a node sends a *LEAVE* message to the AP of the cell it belongs to. If that node was the only member the group in the cell, the AP has to notify the RP. Note that, in contrast to HRPM, no *LOCATION−UPDATE* messages are required in HGMR, and no handoff process, since nodes are static and they do not move to different cells.

When a source has data packets to send, it uses HRPM's unicast-based forwarding strategy to send the packets to each AP along the *Source* → *APs* overlay tree. But within each cell, instead of constructing an *AP* → *Members* overlay tree, HGMR uses GMR's cost over progress optimizing broadcast algorithm to select the next relay nodes at each hop. Adjusting the value for the decomposition index d, we can always ensure that the number of members an AP is responsible for does not grow too large. Hence, the use of GMR within each cell instead of HRPM's unicast-based forwarding strategy helps to reduce the number of transmissions while maintaining a low encoding overhead. HGMR adopts both HRPM's and GMR's policies in dealing with holes in sparse topologies. When routing to a hashed location (RP or AP), HGMR uses HRPM's face routing, while when routing from an AP to a set of group members within a cell, it uses GMR's multicast face routing. Note, however, that holes are expected to be a very rare case in HGMR which targets dense topologies.

The operation of HGMR is shown in Fig. 3(a and b). Figure 3(a) shows the overall picture, which includes the geographic division of the deployment area into cells, the *Source* → *APs* overlay tree (same as the one constructed by HRPM in Fig. 2(c)), and the *AP* → *Members* trees, one within each cell that contains some destinations. These trees are constructed using GMR's localized neighbor selection algorithm, and hence they are not overlay trees, as opposed to HRPM. Figure 3(b) shows in more detail the *AP* → *Members* tree construction in cell 1. Here, the AP node *A* initially selects node *H* as relay node responsible for destination *C*, and node *G* as relay node responsible for destinations *D*, *E*, *F*, and it can also reach directly destination *B*. Hence it sends only one message, with all this information to all its three neighbors, *H*, *G*, and *B*, exploiting the WMA. Node *B* finds that it is not responsible for any other destination, hence it does not rebroadcast the message. Node *H* rebroadcasts it to destination *C*. Node *G* can reach directly destination *F* and it also selects it as a relay for destinations *E* and *D*. *F* can reach *E* directly and it selects node *I* as a relay for destination *D*. Finally, *I* sends the message to *D*.

There is one potential drawback with geographic hashing. Not restricting the hash function (as in the case for HRPM) may result in the RP being very far from the source or the destinations. The problem is not intense in HRPM, which is designed for mobile ad hoc networks, since any such situation is only temporary. But in HGMR, which aims at *static* sensor networks, such a situation could result in unnecessarily high overhead. To alleviate the problem, in HGMR we use a hash function that generates locations within a smaller square region near the center of the whole region, as in GHLS [18]. We set the length of the side of that smaller area to be $\alpha = 0.5$ times the length of the whole deployment area, since [18] showed that this value achieves the best tradeoff between locality and load balancing.

4 Evaluation in an ideal environment

In this section, we evaluate the performance of GMR, HRPM and HGMR in an ideal environment. The goal is to compare how the protocols scale under ideal conditions with no packet loss.

4.1 Methodology

We implemented GMR, HGMR and HRPM in Glomosim [19], a widely used wireless network simulator with a detailed and accurate radio model. To simulate ideal conditions, we used the free space propagation model and modified the MAC layer to remove collisions. Under this setup, all packets multicast by the source are properly received by all the destinations. We implemented HRPM following [9] and GMR following [8]. In all sections HRPM adjusts the decomposition index *d* to the group size based on the equations in [9], using $\omega = 20\% \times$ *PacketSize*. The same *d* is then used for HGMR.

The evaluation consists of two parts. In the first part, we evaluate the scaling properties of the three protocols as a function of the number of multicast members. For this part, we placed 500 static sensor nodes in a 2400 m × 2400 m field and varied the number of multicast members between 25 and 250. In the second part we evaluate the three protocols with respect to the network size. For this experiment, we varied the number of sensor nodes between 100 and 1000, but we kept the same node density (by varying the area of the deployment field), 20 nodes per radio range, and the same member density (30% of the nodes).

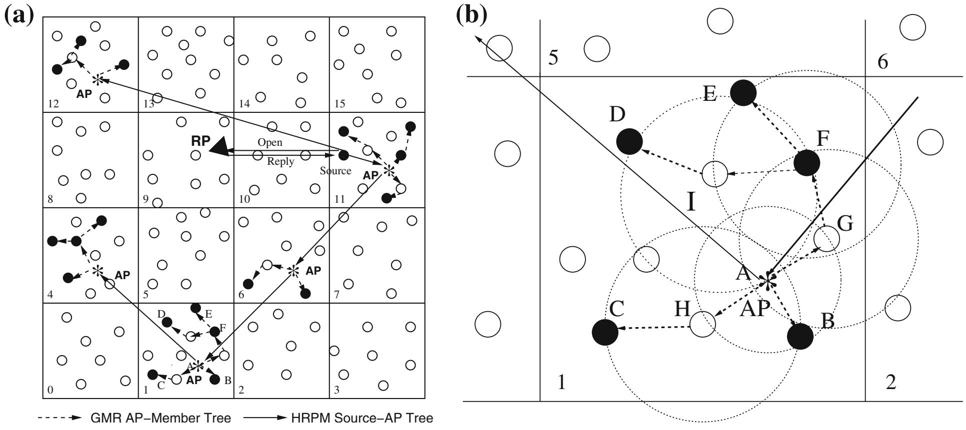


Fig. 3 (**a**) Data delivery in HGMR ($d = 4$). Both the HRPM *Source* → *APs* overlay tree and the GMR *AP* → *Members* trees are shown. The *AP* → *Members* trees connect both members (black dots) and non-members (white dots), as opposed to the overlay trees in Fig. 2(c). (**b**) *AP* → *Members* tree construction in cell 1 using GMR's neighbor selection algorithm

In all cases, the simulation duration was 600 s. We had one multicast source that sent 512-byte packets at a constant rate of 2 packets/s.

4.1.1 *Evaluation metrics*

The following metrics are used to evaluate the efficiency of the three multicast protocols:

- *Number of transmissions*: The total number of transmissions used to deliver the packets from the source to all the destinations. It measures the efficiency of the multicast paths selected. A reduction in the number of transmissions results both in *bandwidth efficiency* (which is always important in wireless networks, where bandwidth is limited), and in *energy efficiency*, which is particularly important in energy-constrained wireless sensor-networks.
- *Percentage of Forwarding Nodes (FNs)*: The number of nodes (including source) that transmitted at least one data packet divided by the total number of nodes.
- *Normalized Encoding Overhead (NEO)*: The ratio of the total number of encoding bytes transmitted at every hop to the total number of data bytes transmitted at every hop. Encoding bytes are the bytes used at each data packet to encode the information required by each protocol.

4.2 Results

4.2.1 *Impact of group size*

We first evaluate the performance of the three protocols under different group sizes. The results are shown in Fig. 4. We make the following observations.

In Fig. 4(a) we observe that HRPM has a much higher number of transmissions compared to HGMR and GMR and the gap increases with the group size. For a group size of 250 members, HRPM has 119% more tranmsissions than HGMR and 138% more transmissions than GMR. GMR uses MAC layer broadcast to transmit packets, exploiting WMA, while HRPM uses MAC layer unicast. Hence at each node part of the multicast tree, only one transmission is required for GMR, and these transmissions can be heard by all 1-hop neighbors. On the other hand, with HRPM a node has to transmit the same packet separately to each of its neighbors selected as forwarding nodes. HGMR uses MAC layer unicast along the *Source* → *APs* and MAC layer broadcast along the *AP* → *Members* tree. By properly adjusting the value of decomposition index *d*, we can limit the number of transmissions in the higher level tree, without increasing the encoding overhead. Hence HGMR's total number of transmissions is only 8% more than GMR's.

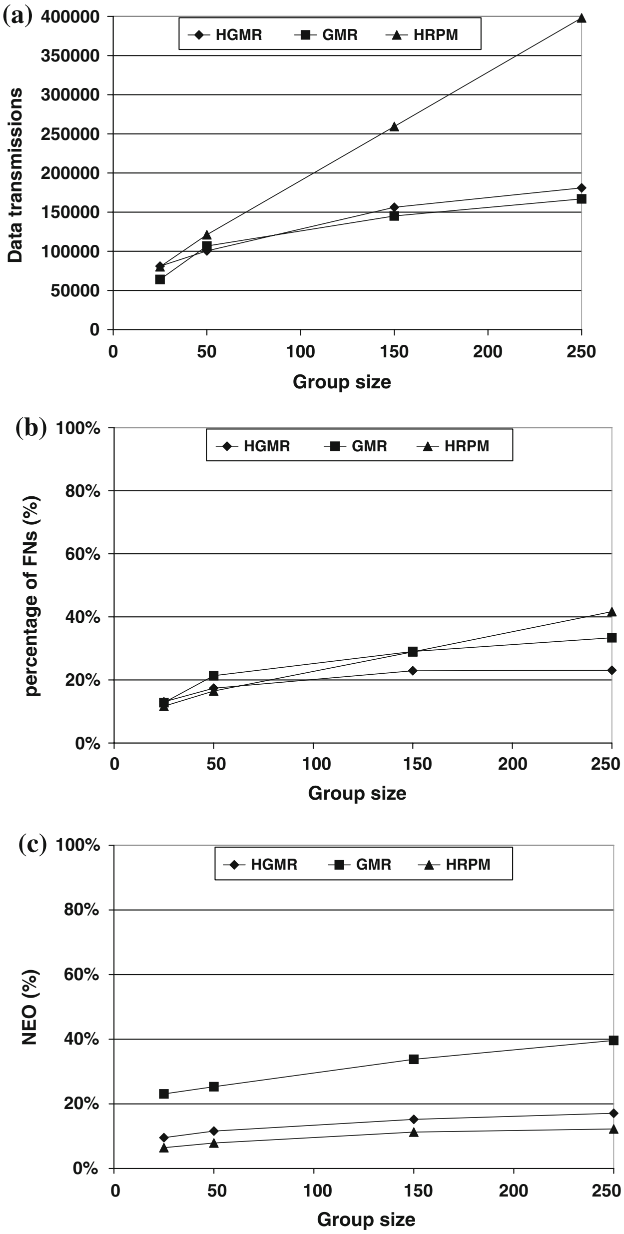


Fig. 4 Performance comparison of GMR, HGMR and HRPM for different group sizes in an ideal environment. (**a**) Data transmissions comparison. (**b**) FN comparison. (**c**) NEO comparison

In Fig. 4(b) we observe that HRPM uses more forwarding nodes than the other two protocols for large group sizes. The percentage of FNs is 42% for HRPM vs. 33% for GMR and 23% for HGMR, with 250 members. HRPM constructs minimum spanning overlay trees to deliver packets to the destinations (the *Source* → *APs* and *AP* → *Members* trees) and uses geographic routing to route packets between any two nodes of an overlay tree. Although this technique is quite efficient for small group sizes (HRPM has the smallest percentage of FNs for up to 50 members), it becomes inefficient for large group sizes, because more intermediate nodes are used as forwarders between two overlay nodes. On the other hand, GMR's forwarding node selection algorithm explicitly tries to minimize the number of FNs at each (real) routing hop. This strategy is proved more efficient, as the group size increases, and hence GMR uses fewer FNs for larger group sizes. Interestingly, HGMR has the smallest percentage of FNs among the three protocols, and most importantly, this percentage is almost stabilized to 23%, while it increases with group size for the other two protocols. This shows that the inefficiency for HRPM with large group sizes comes from the *AP* → *Members* trees, while for GMR the number of transmissions mostly increases due to the first hops around the source—these hops select many forwarders, since they have to reach many members. HGMR uses the most efficient of the two schemes for each of the two trees, and hence it achieves the lowest percentage of FNs.

Figure 4(c) shows the main advantage of HRPM over GMR and HGMR. HRPM was designed with the main goal of keeping the encoding overhead low even for large group sizes and it achieves it by dividing the deployment area in cells and the group into smaller subgroups. In Fig. 4(c) we observe that HRPM's NEO only slightly increases with the group size and it remains less than 12%. On the other hand with GMR a node sends the same data packet to all its neighbors and each packet includes a list of all the neighbors selected as forwarding nodes as well as information about all the destinations the selected neighbors are responsible for. As the group size increases, the source and nodes near the source have to include information about more and more destinations in the data packets they send—in the case of 250 members, the source has to include all the 250 members in the packets it sends. This increases the NEO for GMR up to 40% in case of 250 members. Even for small group sizes (25–50 members), NEO for GMR is 3–4 times higher than for HRPM. Although this increase in NEO has no effect in an ideal environment, it has a severe impact on GMR's performance in a realistic environment, as we will see in Sect. 5. Finally, HGMR uses the same hierarchy as HRPM. For the *Source* → *APs* tree, NEO is same as for HRPM; for the *AP* → *Members* trees, HGMR uses GMR's algorithm, but the number of receivers is kept small (with a proper selection of *d*). Hence the total NEO for HGMR does not increase much compared to HRPM; it only reaches 17% for 250 members.

4.2.2 *Impact of network size*

Next, we evaluate the performance of the three protocols under different network sizes. The results are shown in Fig. 5.

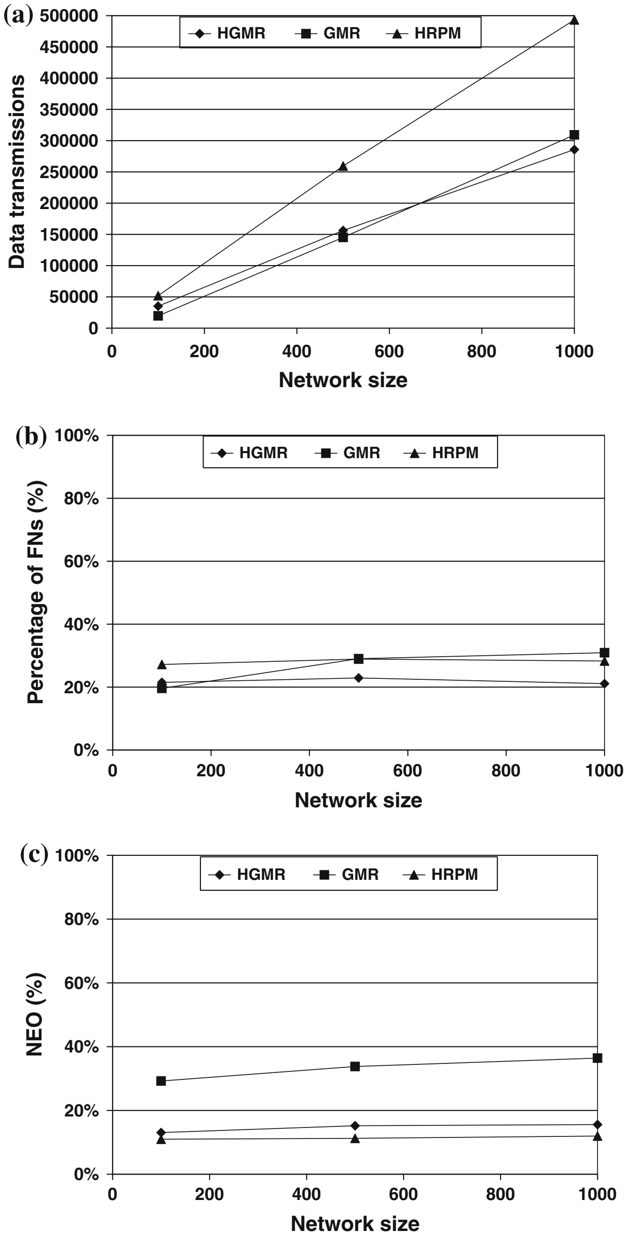


Fig. 5 Performance comparison of GMR, HGMR and HRPM for different network sizes in an ideal environment. (**a**) Data transmissions comparison. (**b**) FN comparison. (**c**) NEO comparison

The key observations are as follows: Fig. 5(a) shows that the number of transmissions increases with the network size for all the three protocols. Again, HRPM has the steepest increase. In the case of 1000 nodes, HRPM incurs 59% more transmissions than GMR and 72% more transmissions than HGMR. Interestingly, for such a large network size GMR is no longer the most efficient protocol in terms of number of transmissions. The large network size increases the lengths of the paths from the source to the destinations and GMR's FN selection algorithm selects more nodes, in trying to increase the progress ratio, thus deteriorating forwarding efficiency. This is also shown in Fig. 5(b), in which we observe that the percentage of FNs remains constant for the two hierarchical protocols, HRPM and HGMR, but it increases with the network size for GMR. HGMR again tries to balance the size of the *Source* → *APs* and *AP* → *Members* trees, and hence for *AP* → *Members* trees, GMR's algorithm is never applied to a very large tree and this reduces the number of transmissions for HGMR compared to GMR.

Finally, from Fig. 5(c) we observe that NEO increases with the network size for GMR, while for HRPM and HGMR it remains unaffected and much lower compared to GMR. With 1000 nodes, NEO is 12% for HRPM and 15% for HGMR but it reaches 36% for GMR. By increasing the network size, the number of multicast members also increases with a constant member density, and this causes GMR's encoding overhead to increase, as we explained in case of Fig. 4(c). HRPM and HGMR on the other hand can adjust the number of members within each subgroup by varying the parameter d and hence, they maintain a low encoding overhead for all network sizes.

In summary, we saw that HRPM incurs too many packet transmissions, when the group size or the network size increases, and it also involves many nodes in packet forwarding. Hence, HRPM is not appropriate for sensor networks, where energy conservation is of great importance. On the other hand, GMR incurs much fewer packet transmissions compared to HRPM, and it also uses fewer FNs, by exploiting WMA. However, for very large network sizes, the greedy neighbor selection of GMR is not as efficient and the percentage of FNs increases, thus increasing energy consumption and limiting network's lifetime. In addition, GMR has much higher encoding overhead compared to HRPM, and this will affect its performance in realistic scenarios. HGMR, which is a hybrid of the two protocols, combines the high forwarding efficiency of GMR with the low encoding overhead of HRPM and it can scale very well, with respect to both the group and the network size.

5 Evaluation in a realistic environment

In this section, we evaluate the performance of GMR, HRPM and HGMR in a realistic environment.

5.1 Methodology

To simulate a realistic environment, in Glomosim we used an IEEE 802.11 radio with a bit rate of 2 Mbps and a transmission range of 250 m. The *TwoRay* propagation model was used instead of the *Free Space*. Under this setup, packet loss can happen due to two reasons: (i) Collisions may happen when two or more nodes transmit a packet at the same time. The probability of collisions increases with the number of forwarding nodes and the packet size. (ii) Packets may be corrupted due to noise or the receiver may be unable to decode them due to low SNR. The probability of packet corruption increases with the packet size.

We followed the same evaluation methodology as in Sect. 4, varying again the number of multicast members and the network size. All other simulation parameters remain the same as in Sect. 4.

5.1.1 *Evaluation metrics*

In addition to the metrics used in Sect. 4, we used the following evaluation metrics:

- *Average Packet Delivery Ratio (PDR)*: The number of the data packets delivered to a multicast group member divided by the number of data packets transmitted by the source, averaged over all multicast group members. This metric is necessary, since in a realistic environment, there is packet loss, as we mentioned above.
- *Average Delivery Latency (Delay)*: Packet delivery latency averaged over all multicast packets delivered to all receivers. It includes all possible delays caused by queuing at the interface queues, backoff at the MAC layer when the channel is busy, as well as propagation and transfer times.
- *Forwarding Cost (FC)*: The total number of data packet transmissions divided by the total number of packets received by all the multicast members. It gives the average number of transmissions required per delivered packet. In an ideal environment, the number of data received (denominator) is same for all protocols, and hence this metric degenerates to be the same as the total number of transmissions. In a realistic environment, the PDR is different for each protocol, and hence this metric combined with the total number of transmissions gives a better picture of the forwarding efficiency of each protocol.

5.2 Results

5.2.1 *Impact of group size*

Figure 6 shows the performance of the three protocols under different group sizes.

Figure 6(a and b) show that HGMR outperforms the other two protocols in terms of both PDR and packet delivery latency. We observe that PDR is less than 100% for all three protocols, since packet loss occurs due to bit

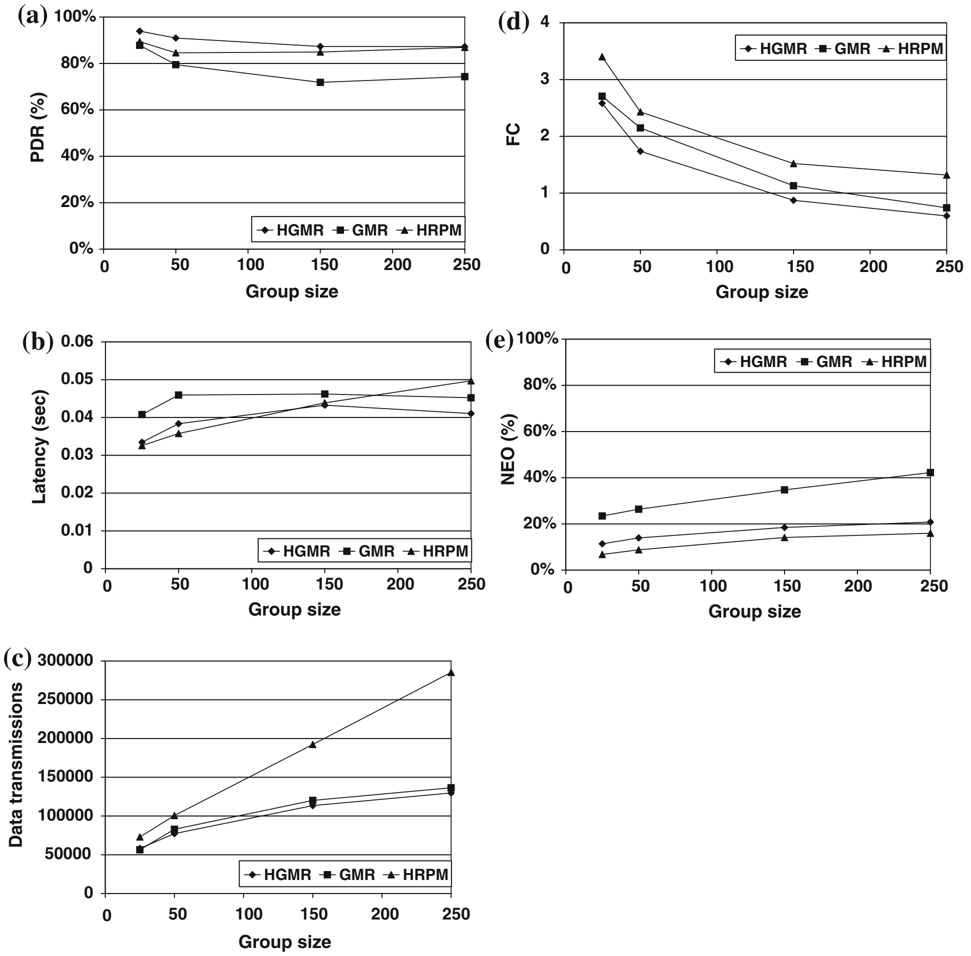


Fig. 6 Performance comparison of GMR, HGMR and HRPM for different group sizes in a realistic environment. (**a**) PDR comparison. (**b**) Delivery latency comparison. (**c**) Data transmissions comparison. (**d**) FC comparison. (**e**) NEO comparison

errors and due to collisions. However, HGMR maintains a high PDR above 87% even for large group sizes and a packet delivery latency between 33 and 43 ms. On the other hand, performance of GMR suffers for large group sizes. PDR can become as low as 74% and delivery latency varies between 40 and 46 ms, always larger than with HGMR. Finally, HRPM performs almost as well as HGMR in terms of PDR; especially, as the group size increases, PDR becomes almost the same for the two protocols. The delivery latency is also similar for the two protocols, except for the case of very large group size (250 members); in this case the delivery latency for HRPM increases and becomes higher even compared to GMR.

This difference in terms of PDR and delivery latency among the three protocols can be explained if we look at the rest of our evaluation metrics in Fig. 6(c–e). Figure 6(c) shows that the total number of transmissions is much lower for GMR and HGMR compared to HRPM, which is consistent with our observation in Fig. 4(a) for the ideal case. The difference increases with the group size and it reaches up to 120% for a group size of 250 members. Note also that the total number of transmissions is lower for all three protocols, compared to the ideal case in Fig. 4(a), due to packet losses. We have also seen in Fig. 4(b) that HRPM uses many more nodes as forwarders compared to the other two protocols, which explicitly try to minimize the number of forwarders through the cost over progress ratio they use in their neighbor selection algorithm—the number of FNs remains the same under ideal or realistic conditions.

Finally, Fig. 6(d) shows the Forwarding Cost (FC) for the three protocols. FC combines the number of transmissions and the number of packets received by all members, hence it gives a better idea of the forwarding efficiency of a protocol in a realistic environment. First, we observe that FC decreases with the group size for all three protocols, since the denominator (number of total packets received by all members) is increased. Similar to the other two efficiency metrics, FC also remains higher for HRPM compared to GMR and HGMR; for a group size of 250 members, FC with HRPM is equal to 1.32, while it is equal to 0.74 for GMR and 0.6 for HGMR. Note that for large group sizes, FC for GMR and HGMR is less than 1, due to multicast advantage. The same observation was made in [9] for ODMRP, a popular broadcast-based multicast protocol. This means that a single transmission with GMR and HGMR is enough for more than one receiver.

In spite of being wasteful in terms of network resources, HRPM achieves higher performance (higher PDR and lower delivery latency) compared to GMR and very close to HGMR. This is explained by the Normalized Encoding Overhead (NEO), which is shown in Fig. 6(e) for the three protocols. In this figure we observe that HRPM has the lowest NEO among the three protocols (NEO can reach up to 16% for HRPM, up to 21% for HGMR and up to 43% for GMR, for all group sizes), which is again consistent with Fig. 4(c) for the ideal case. On the other hand, NEO for GMR is much higher than for the other two protocols and it increases rapidly with the group size. With 250 members, GMR on average uses 43 encoding bytes for every 100 data bytes transmitted.

This increase to the packet size increases both the probability of bit errors (which causes packet corruption) and the probability of packet collisions. Hence with a similar number of transmissions as HGMR, GMR delivers fewer packets. On the other hand, HRPM transmits many more packets, but these packets are much smaller, compared to GMR’s packets. The increase of the probability of collisions in HRPM due to the large number of transmissions is counterbalanced by the decrease due to the small packet size. Only with a very large group size we can see some performance degradation for HRPM, not in terms of PDR but in terms of delivery latency, which increases due to increased contention among the large number of packets transmitted by the protocol. Finally HGMR maintains a much lower number of transmissions compared to HRPM and a much lower NEO compared to GMR, and by balancing these two factors, it can achieve the highest performance and the best network resource conservation among the three protocols.

5.2.2 *Impact of network size*

Figure 7 shows the performance of the three protocols under different network sizes. The main observations are as follows.

In Fig. 7(a) we observe that PDR drops with network size for all three protocols. But when the network size changes from 500 to 1000 nodes, this drop is negligible for HRPM and HGMR (from 84 to 82%), but significant for GMR (from 72 to 60%). Similarly, in Fig. 7(b) we observe that for a 1000-node network delivery latency for GMR is much higher compared to the other two protocols (68 ms for GMR vs. 54 ms for HRPM and 53 ms for HGMR). This shows that GMR does not scale well for very large network sizes. On the other hand, the performance of the other two protocols, HGMR and HRPM is almost identical; HGMR only slightly outperforms HRPM.

If we look at Fig. 7(c), we will see that again HRPM incurs a much higher number of transmissions compared to the other two protocols, and the gap increases with the network size—it reaches 88% for 1000 nodes. On the other hand, GMR and HGMR have similar number of transmissions. However, GMR has lower PDR and higher delivery latency compared to the other two protocols, due to the large packet size. In Fig. 7(e) we observe that NEO for GMR becomes larger than 40% in a 1000-node network. As we mentioned before, an increase in the packet size increases both the probability of bit errors and the probability of collisions. The later is also exacerbated by the large number of FNs (in Fig. 5(b) we saw that the percentage of FNs increases with the network size for GMR while it remains constant for the other two protocols). This results in packet loss which reduces the PDR. Also, contention among neighboring nodes increases, and this increases packet delivery latency. For HRPM, the probability of collisions should also be increased, in this case due to the large number of transmissions. But the low NEO in Fig. 7(e) shows that the packet size is kept small for HRPM, hence the probability of collisions finally remains low, as well as the probability of bit errors.

Figure 7(d) shows that forwarding cost drops with network size for HGMR and HRPM, since the total number of received packets is increased, as we also explained in the case of large group size. HGMR is again the most efficient protocol, maintaining a FC lower than 1 for all network sizes, due to the WMA. Interestingly enough, GMR does not follow the trend of the other two protocols, and its FC increases with the network size, and it is always more than 1, although it also exploits WMA, in a higher degree than HGMR. However, PDR drops significantly for GMR, as we show in Fig. 7(a), and this reduction in PDR affects FC, which finally becomes higher than 1.

6 Evaluation with a non-uniform group member distribution

In Sects. 4 and 5 we considered a uniform group member distribution. Although we believe this to be a reasonable assumption for large, dense, application-oriented sensor networks, in this section we evaluate HGMR’s performance under non-uniform member distributions. We consider a 2400 m × 2400 m deployment field with 1000 sensors and 75 group members. The 1000 nodes are uniformly randomly placed in the field, similar to in Sects. 4 and 5. However, for the member placement, we consider two extreme cases of non-uniform distribution. In *Case* 1, 39 group members are selected among the nodes in the upper right square with dimensions 400 m × 400 m, and

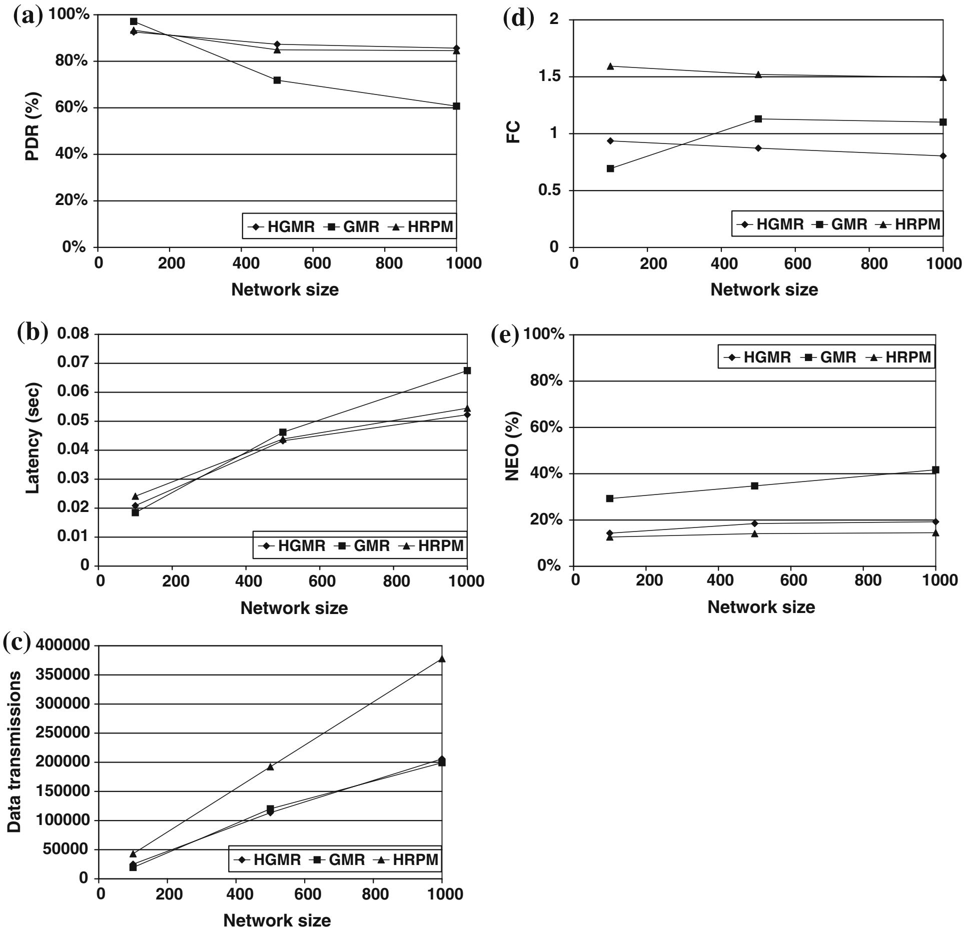


Fig. 7 Performance comparison of GMR, HGMR and HRPM for different network sizes in a realistic environment. (**a**) PDR comparison. (**b**) Delivery latency comparison. (**c**) Data transmissions comparison. (**d**) FC comparison. (**e**) NEO comparison

the remaining 36 in the lower left square with the same dimensions. In *Case* 2, we consider an even more extreme situation, and all 75 members are selected among the nodes in the upper right square with dimensions 400 m × 400 m.

We performed the simulations with the realistic setting described in Sect. 5, The results are shown in Fig. 8.

In Fig. 8(a and b), we observe that HGMR outperforms HRPM and GMR even in extreme situations of non-uniform group member distributions. In both cases, HGMR achieves a PDR almost identical to HRPM, and much higher than GMR (39 and 60% higher for the two cases). In terms of delivery latency, HGMR significantly outperforms both GMR and HRPM in both cases. The improvement is 46 and 54%, for *Case* 1 and *Case* 2, respectively, over GMR, and 53%, and 73%, respectively, over HRPM.

Figure 8(c–f) explain the performance difference in Fig. 8(a and b). In Fig. 8(c), we observe that the use of unicast in HRPM makes the protocol very inefficient in terms of FNs. HRPM constructs a very dense overlay tree, and the total number of FNs is 76 in *Case* 1 and 66 in *Case* 2. On the other hand, HGMR exploits the broadcast advantage and the fact that group members are gathered in a small area with very high density, and uses only 51 FNs in *Case* 1, and only 29 FNs in *Case* 2. Surprisingly, GMR also uses too many FNs, although it is also broadcast-based. In *Case* 1 the number of FNs with GMR is 85, the highest among the three protocols; in *Case* 2 the number of FNs with GMR is 53, lower than with HRPM but still much higher compared to HGMR. The reason is that GMR does not use a hierarchy, in contrast to the other two protocols. The localized neighbor selection algorithm of GMR, although efficient when members are uniformly deployed, fails in case of non-uniform distributions, since trying to separately optimize the cost-over progress ratio for many forwarders close to each other, results in too many FNs, as the packets travel towards the only one or two cells that contain the group members. On the other hand, the two hierarchical protocols construct an overlay tree with only 2 or 1 branches in the two scenarios we examine here, and forward the data to the cells that contain members using only a few FNs. Within these cells, HRPM builds new overlay trees and unicasts packets from the APs to each individual member, which results in too many FNs. On the other hand, HGMR uses here GMR's neighbor selection algorithm, which is very efficient for dense, uniform topologies, and delivers packets to members using very few FNs.

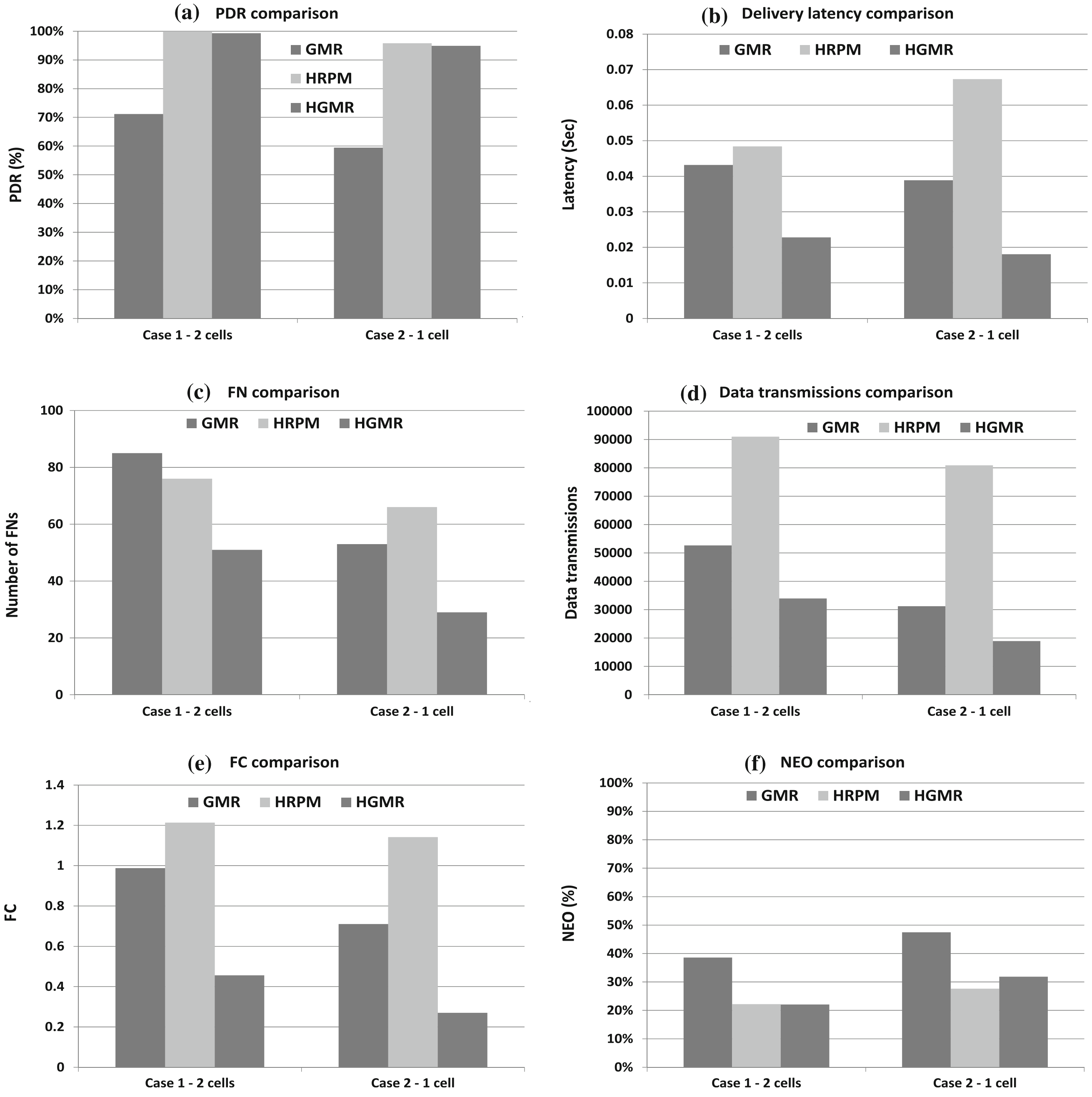


Fig. 8 Performance comparison of GMR, HGMR and HRPM for two different non-uniform group member distributions. *Case 1*: members are gathered in the upper right and lower left corner of the field. *Case 2*: members are gathered in the upper right corner of the field

The large number of FNs and the use of unicast make HRPM very inefficient in terms of transmissions. HRPM incurs the largest number of transmissions among the three protocols, as shown in Fig. 8(d), and it has the highest FC, as shown in Fig. 8(e)—actually HRPM is the only protocol with an FC higher than 1. This large number of transmissions increases the amount of congestion in the network, and results in significant packet losses due to collisions. The built-in reliability mechanism of 802.11 unicast masks the majority of these losses, however it increases the latency, as we saw in Fig. 8(b). GMR exploits broadcast, thus it incurs fewer transmissions compared to HRPM, and it also has a lower FC than HRPM, but still higher than HGMR. Finally, with HGMR, packets are transmitted using unicast in the largest part of their route, from the RP to the APs, which offers increased reliability, but they are broadcast within the two (or one) cells, from the AP to the members. Because of the large density of members in the cells, a few broadcast transmissions are enough to deliver each data to most of the members. Hence, HGMR incurs the lowest number of transmissions among the three protocols, 36 and 39% fewer than GMR, and 63 and 77% fewer than HRPM, for *Case* 1 and *Case* 2, respectively. It also has the lowest FC among the three protocols: FC is only 0.46 for HGMR, 0.98 for GMR, and 1.21 for HRPM in *Case* 1, and 0.27 for HGMR, 0.71 for GMR, and 1.14 for HRPM in *Case* 2.

Finally, Fig. 8(f) compares the NEO for the three protocols. This figure shows again the inefficiency of GMR for large groups. Since all group members are concentrated in a small area, GMR has to include a large number of group member IDs and locations in the headers of data packets, as they travel from the source to the group member area(s). In the most extreme situation, i.e., in *Case* 2, packets travel all the way to the upper right square containing information about all 75 members in their header. This results in a high NEO for GMR, equal to 39% in *Case* 1, and 47% in *Case* 2. With the two hierarchical protocols, when the packets travel from the RP to the APs, the encoding overhead is very low and constant, equal to $d^2/8$ bytes, and this reduces the total NEO for both protocols significantly, compared to the flat GMR. The two protocols achieve the same NEO, equal to 22%, in *Case* 1, and HRPM slightly outperforms HGMR (28% vs. 31%) in *Case* 2. The reason is that in *Case* 2 HGMR still has to encode all 75 members in the packets broadcast from the AP, while HRPM distributes this overhead over several branches of the unicast overlay $AP \rightarrow Members$ tree.

7 Related work

There has been a large amount of work on efficient multicast in wireless ad hoc networks. In addition to GMR and HRPM, HGMR is closely related to previous location-based multicast protocols and hierarchical non-location-based multicast protocols.

7.1 Location-based multicast protocols

Previous location-based protocols [1–3] were proposed for small groups due to the constraint of encoding either the entire tree or the destinations in the data packet headers. In DSM [1], each node floods its location in the network. DSM constructs a physical Steiner tree using the TM heuristic [20] at the source, optimally encodes the physical multicast tree into each packet, and delivers the packet using source routing. LGT [2] requires each group member to know every other group member's location. LGT proposes two overlay multicast trees: a bandwidth-minimizing LGS tree and a delay-minimizing LGK tree. PBM [3] does not explicitly construct trees but rather relies on a multicast geographic forwarding strategy similar to the hop-by-hop forwarding proposed by SGM [21] and DDM [22]. MgCast [23] sends multicast data to a geographical area rather than multiple destinations. It uses position information to build a multicast tree that tries to minimize the number of links. However, as the wireless medium is characterizes by its broadcast nature, the cost of a tree is better characterized by the number of nodes rather than the number of links.

The SPBM protocol [24] shares with HRPM the essence of improving the scalability of location-based multicast using hierarchical group management. A fundamental difference between the two is that SPBM uses *flooding* in hierarchical group management, while HRPM uses *mobile geographic hashing* (convergence to the rendezvous point) in hierarchical group management which does not incur any flooding cost.

7.2 Hierarchical non-location-based multicast protocols

Several hierarchical non-location-based protocols have been proposed which can be overlay or non-overlay based. Protocols such as AMRIS [28] and PAST-DM [29] propose an overlay-based approach in which the overlays are a form of hierarchies.

An example of a non-overlay hierarchical multicast protocol is HDDM [30] which extends DDM to include a hierarchical structure. Similar to HDDM, HGMR and HRPM also leverage the well known technique of introducing a hierarchical structure to reduce overhead. Despite this similarity, HDDM is a topology-aware approach while HGMR and HRPM are location-aware approaches, and hence the design challenges and issues are very different. In particular, HGMR/HRPM provide location management and routes using locations rather than topology.

The work in [31] proposed the use of *cores* to reduce control traffic for creating multicast delivery structures. They propose that group members form a multicast group by sending join requests to a set of cores. Rendezvous points are similar in concept to core nodes. However RPs/APs in HGMR/HRPM can be located without any overhead using geographic hashing and can be more resilient to mobility due to not being tied to a particular node whose movement needs to be tracked.

8 Conclusions and future work

In this paper, we have presented HGMR, a new location-based multicast protocol for wireless sensor networks. HGMR seamlessly incorporates innovations in location-based multicast and optimizes them for wireless sensor networks by *simultaneously* providing both *energy-efficiency* as well as *scalability* to large networks. Our simulation studies confirm that HGMR combines the strengths of the two protocols it leverages (HRPM and GMR): In an ideal environment, HGMR incurs a number of transmissions either very close to or lower than GMR, and an encoding overhead very close to HRPM, as the group size or the network size increases. In a realistic environment, HGMR achieves a PDR close to that of HRPM and much higher than GMR, and the lowest packet delivery latency among the three protocols. Finally, HGMR maintains its superior performance compared to the other two protocols even under non-uniform group member distributions.

In a more general context, the design of HGMR provides a framework for simultaneously handling scalability and different aspects of energy efficiency in location-based multicast for wireless sensor networks. The use of geographic hashing achieves scalability by keeping the per-packet encoding overhead constant at virtually no maintenance cost. Within each cell at each level in the hierarchy, different metrics related to energy efficiency can be optimized, by selecting an appropriate protocol for delivering data from the AP to the members of that cell. In this paper we used GMR for data forwarding within each cell of the hierarchy in order to reduce the total number of transmissions, which reduces both total energy consumption and bandwidth usage. As another example, we could optimize the balanced energy consumption instead of the total energy consumption, by replacing GMR with a protocol that rotates between alternative forwarding trees within each cell. A simple way to achieve that is to periodically change the AP within each cell, by periodically adjusting the hash function used for selecting APs, and using the existing multicast structure to distribute the new function to the multicast receivers. Also, sleep-wake MAC protocols for sensor networks that periodically put some nodes to sleep can be used within each cell to conserve energy. In our future work, we plan to extend the HGMR framework to support such different cost functions, as well as to provide manycast and anycast services.

Acknowledgments This work is partially supported in part by NSF grant ANI-0338856, NSERC Strategic Grant STPGP 336406, and the UK Royal Society Wolfson Research Merit Award.

References

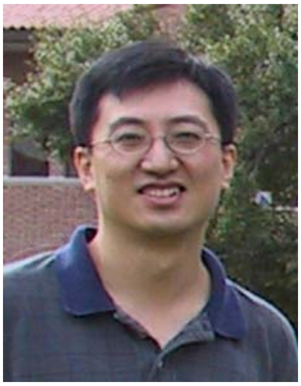
1. Basagni, S., Chlamtac, I., & Syrotiuk, V. (2001). Location aware, dependable multicast for mobile ad hoc networks. *Computer Networks, 36*, 659–670.
2. Chen, K., & Nahrstedt, K. (2002). Effective location-guided tree construction algorithms for small group multicast in MANET. In *Proceedings of the IEEE INFOCOM*, June 2002.
3. Mauve, M., Fuessler, H., Widmer, J., & Lang, T. (2003). *Position-based multicast routing for mobile ad-hoc networks*, Tech. Rep. CS TR-03-004. University of Mannheim.
4. USCG Navigation Conter GPS page, http://www.navcen.uscg.nil/gps/default.html. January 2000 [Online]. Available: http://www.navcen.uscg.nil/gps/default.html.
5. Royer, E. M., & Perkins, C. E. (1999). Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Proceedings of the MobiCom*, August 1999.
6. Jetcheva, J. G., & Johnson, D. B. (2001). Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *Proceedings of the ACM MobiHoc*, October 2001.
7. Lee, S.-J., Gerla, M., & Chiang, C.-C. (1999). On-demand multicast routing protocol. In *Proceedings of the IEEE WCNC*, September, 1999.
8. Sanchez, J., Ruiz, P., Liu, X., & Stojmenovic, I. (2006). GMR: Geographic multicast routing for wireless sensor networks. In *Proceedings of the IEEE SECON*.
9. Das, S. M., Pucha, H., & Hu, Y. C. (2007). Distributed hashing for scalable multicast in wireless ad hoc networks. *IEEE TPDS, 47*(4), 445–487
10. Ye, F., Zhong, G., Lu, S., & Zhang, L. (2003). Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the IEEE ICDCS*.
11. Schurgers, C., Tsiatsis, V., Ganeriwal, S., & Srivastava, M. B. (2002). Stem: Topology management for energy efficient sensor networks. *IEEE Transactions on Mobile Computing, 1*(1).
12. Xu, Y., Heidemann, J., & Estrin, D. (2001). Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking (MobiCom)*, July 2001.
13. Chen, B., Jamieson, K., Balakrishnan, H., & Morris, R. (2001). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking (MobiCom)*, July 2001.
14. Lee, S.-J., Su, W., & Gerla, M. (2002). On-demand multicast routing protocol in multihop wireless mobile networks. *Mobile Networks and Applications, 7*(6), 441–453.
15. Bose, P., Morin, P., Stojmenovic, I., & Urrutia, J. (1999). Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the ACM DialM Workshop*, August 1999.
16. Karp, B., & Kung, H. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM MobiCom*, August 2000.
17. Kuhn, F., Wattenhofer, R., & Zollinger, A. (2003). Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proceedings of the ACM MobiHoc*, June 2003.
18. Das, S. M., Pucha, H., & Hu, Y. C. (2005). Performance comparison of scalable location services for geographic ad hoc routing. In *Proceedings of the IEEE INFOCOM*, March 2005.
19. Zeng, X., Bagrodia, R., & Gerla, M. (1998). Glomosim: A library for parallel simulation of large-scale wireless networks. In *Proceedings of the PADS Workshop*, May 1998.
20. Takahashi, H., & Matsuyama, A. (1980). An approximate solution for the steiner problem in graphs. *Mathematica Japonica, 24*, 573–577.
21. Boivie, R., Feldman, N., & Metz, C. (2000). Small group multicast: A new solution for multicasting on the internet. *IEEE Internet Computing, 4*(3), 75–79.

22. Ji, L., & Corson, S. (2001). Differential destination multicast–A MANET multicast routing protocol for small groups. In *Proceedings of the IEEE INFOCOM*, April 2001.
23. Mizumoto, A., Yamaguchi, H., & Taniguchi, K. (2004). Cost-conscious geographic multicast on MANET. In *Proceedings of the IEEE SECON*.
24. Transier, M., Fuessler, H., Widmer, J., Mauve, M., & Effelsberg, W. (2004). Scalable position-based multicast for mobile ad-hoc networks. In *Proceedings of the First International Workshop on Broadband Wireless Multimedia: Algorithms, Architectures and Applications (BroadWim)*, October 2004.
25. Ruiz, P. M., & Stojmenovic, I. (2006). Cost-efficient multicast routing in ad hoc and sensor networks. In T. Gonzalez (Ed.), *Handbook on approximation algrorithms and metaheuristics*. Chapman and Hall/CRC.
26. Stojmenovic, I. (2006). Localized network layer protocols in sensor networks based on optimizing cost over progress ratio. *IEEE Network, 20*(1).
27. Wu, S., & Candan, K. S. (2006). GMP: Distributed geographic multicast routing in wireless sensor networks. In *Proceedings of the IEEE ICDCS*.
28. Wu, C., & Tay, Y. (1999). AMRIS: A multicast protocol for ad hoc wireless networks. In *Proceedings of the MILCOM*, November 1999.
29. Gui, C., & Mohapatra, P. (2003). Efficient overlay multicast for mobile ad hoc networks. In *Proceedings of the IEEE WCNC*, March 2003.
30. Gui, C., & Mohapatra, P. (2004). Scalable multicasting for mobile ad hoc networks. In *Proceedings of the IEEE INFOCOM*, March 2004.
31. Madruga, E., & Garcia-Luna-Aceves, J. (2001). Scalable multicasting: The core assisted mesh protocol. *ACM/Baltzer Mobile Networks and Applications, Special Issue on Management of Mobility in Distributed Systems, 6*(1).

Author Biographies

Dimitrios Koutsonikolas is currently a Ph.D. student in the School of Electrical and Computer Engineering at Purdue University, USA. Previously, he received a B.Engg. degree from the National Technical University of Athens (NTUA), Greece. His research interests include wireless ad hoc, mesh and sensor networks. He is a recipient of the Ross Fellowship, Purdue University, 2004, and a Tellabs Fellowship, Center for Wireless Systems and Applications (CWSA), Purdue University, 2006.

Saumitra M. Das is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University, USA. Previously, he received a M.S. degree from Carnegie Mellon University, USA and a B.Engg. degree from the University of Bombay, India. His research interests include cross-layer system design for multi-hop wireless networks, scalable routing strategies in wireless ad hoc networks, and mobile robotics.



Y. Charlie Hu is an Associate Professor of Electrical and Computer Engineering and Computer Science at Purdue University. He received his M.S. and M.Phil. degrees from Yale University in 1992 and his Ph.D. degree in Computer Science from Harvard University in 1997. From 1997 to 2001, he was a research scientist at Rice University. His research interests include operating systems, distributed systems, wireless networking, and parallel computing. He has published over 100 papers in these areas. He received the NSF CAREER Award in 2003. He served as a TPC Vice Chair for ICDCS 2007 and the 2004 International Conference on Parallel Processing and a co-founder and TPC co-chair for the International Workshop on Mobile Peer-to-Peer Computing. He is a member of USENIX and a senior member of ACM and IEEE.

Ivan Stojmenovic received a Ph.D. degree in Mathematics. He has held positions in Serbia, Japan, the United States, Canada, France and Mexico. He has published over 200 different papers and edited three books on wireless, ad hoc, and sensor networks with Wiley and IEEE Press. He is currently an editor of several journals including IEEE Transactions on Parallel and Distributed Systems. He has recently guest edited special issues in several journals including IEEE Computer (February 2004), IEEE Network (July 2004), and Wireless Communications and Mobile Computing (Wiley).