2011 IFIP Ninth International Conference on Embedded and Ubiquitous Computing

Energy Efficient and Robust Multicast Routing for Large Scale Sensor Networks

Myounggyu Won and Radu Stoleru
Department of Computer Science and Engineering, Texas A&M University
{*mgwon, stoleru*}*@cse.tamu.edu*

***Abstract*—In this paper we present RE^{2}MR, an energy efficient and robust multicast routing protocol suitable for large scale real-world WSN deployments. RE^{2}MR, a hybrid multicast protocol, builds on the strengths of existing topology-based, hierarchical and geographic multicast solutions, and addresses their limitations. RE^{2}MR establishes a network topology in which multicast member nodes are connected to the root node via near-optimal multicast routing paths. RE^{2}MR discovers deployment area irregularities (e.g., holes) that affect the optimality of multicast routing and considers them when recomputing the near-optimal solution. RE^{2}MR incurs little computational overhead on forwarding nodes, a negligible communication overhead and ensures reliable multicast packet delivery. We implement RE^{2}MR in TinyOS and evaluate it extensively using TOSSIM. RE^{2}MR reduces the energy consumption by up to 57% and the end-to-end delay by up to 8%, when compared with state of art solutions.**

***Keywords*-wireless sensor networks; multicast routing; energy efficiency**

I. Introduction

Multicast is an essential component in many Wireless Sensor Network (WSN) applications. Targeted queries, code updates, and mission assignments are well known examples of multicast services. Unfortunately, traditional tree-based [1] and mesh-based [2] multicast protocols, mainly designed for mobile ad hoc networks (MANET), are not suitable to WSNs. Such topology-based protocols require periodic flooding of control messages to maintain the underlying overlay structure up-to-date, thereby causing the early depletion of energy. Additionally, forwarding nodes have to maintain a routing table for each multicast group. Maintaining, possibly large, state information on a sensor node with limited storage capabilities, is an impractical design decision for WSNs, especially in large scale WSN deployments.

Location-based multicast routing protocols, usually referred as Geographic Multicast Routing (GMR), were proposed as a suitable multicast solution for resource constrained WSNs [3][4][5]. In GMR, the locations of all the subscribers in the multicast group are encoded in each multicast packet, so that the routing decision (e.g., whether the path needs to be split or not) is made on the fly, instead of constructing and maintaining the global tree/mesh routing structure. Although GMR addresses the issues encountered by the topology-based multicast protocols, GMR is not a viable solution for large scale WSN deployments in real environments for several reasons. First, in GMR, the packet header size grows significantly as the size of the multicast group increases. Second, GMR incurs significant computational overhead in forwarding nodes. For example, a forwarding node must compute a heuristic Euclidean Steiner tree [5] or it needs to consider all possible subsets of multicast member nodes [3]–an exponential increase in the computational complexity, with an increase in the multicast group size.

Recently, hierarchical GMR has been proposed as a solution to address the limitations of GMR [6][7][8]. The main idea is to geographically decompose a network into small cells. A *leader* in each cell manages the *subscribers* in that cell. This hierarchical protocol design allows the header size of a multicast packet to be limited. However, the limited packet header size comes at the cost of communication overhead. The control packets are needed for electing a leader in each cell and for managing the local group membership in a cell. Most importantly, the simple network partition into a set of cells results in sub-optimal routing paths from the root node to multicast group member nodes.

To address the aforementioned limitations of the state of art multicast routing solutions, we propose the Robust and Energy-Efficient Multicast Routing (RE^{2}MR) protocol. RE^{2}MR is a hybrid multicast protocol that combines the strengths of the topology-based, geographic and hierarchical multicast solutions. RE^{2}MR, using a solver for the Capacitated Concentrator Location Problem (CCLP), computes the multicast topology that minimizes the sum of path lengths from the multicast root node, to multicast members. To account for realistic WSN deployments, where holes might be present, RE^{2}MR implements a Trajectory-based Lightweight Hole Detection (TLHD) scheme. TLHD is lightweight since it piggybacks on the regular multicast communication, and efficient since it provides the compact representation of a hole. The information on a hole, discovered by TLHD, coupled with an iterative application of CCLP, enable RE^{2}MR to refine the multicast topology towards near-optimality. RE^{2}MR improves its energy efficiency by leveraging the broadcast nature of the wireless medium and by a careful packet header design. Additionally, the multicast topology that RE^{2}MR produces is ideally suited for implementing reliable multicast packet delivery, through fast and efficient recovery from packet loss.

The contributions of this paper are as follows:

978-0-7695-4552-3/11 $26.00 © 2011 IEEE
DOI 10.1109/EUC.2011.33

- We present a hybrid multicast protocol that provides near-optimal path length, low communication and computational overhead, and ensures reliable packet delivery.
- As a component of our multicast protocol, we develop a lightweight hole detection scheme. This scheme can enhance the functionality of other WSN applications in realistic environments.
- We develop a packet forwarding scheme that increases the energy efficiency of our proposed solution and can be integrated with other multicast routing protocols.
- We demonstrate the expected performance gains through theoretical analysis. In addition, through extensive simulations, we show that our protocol outperforms state of art solutions.

II. State of Art

Conventional multicast protocols can be largely categorized into the tree-based and mesh-based protocols. The tree-based multicast protocols [1][9] build a tree structure, either proactively or reactively, to efficiently deliver a packet to subscribers along this tree. The mesh-based protocols [2][10], to better cope with link failures, build a mesh overlay instead. These topology-based protocols incur overhead for constructing the overlay structures and for maintaining the state information about the overlay structure in each node.

Stateless multicast protocols [3][4][5], based on the locations of nodes, do not require the construction and maintenance of underlying global structures like a tree or a mesh. In these protocols, the locations of subscribers are encoded in a packet. Using this location information, the decision on wether a path needs to be split or not is made on the fly, instead of relying on the global structures. However, these protocols are not scalable, because the packet size grows significantly as the number of subscribers increases. Furthermore, these protocols require high computational overhead in each forwarding node to find the optimal subset of neighbors to forward the packet.

To reduce the packet size overhead and ultimately achieve scalability, several hierarchical geographic multicast protocols have been proposed [6][7][8]. In these protocols, a network is divided into a set of cells. In each cell, a specially designated node is elected for managing the group membership. Instead of sending a packet to all the subscribers, a source sends a packet to the leaders, and the leaders distribute the packet to its members. This way the packet header size is reduced, because the header contains only the locations of the leaders. However, these protocols incur additional message and computational overhead for electing the leader and for managing the subscribers in a cell. Most importantly, a simple clustering into a group of cells results in sub-optimal path length.

Recent research, mostly related to RE^{2}MR, mitigates the issues faced by different classes of multicast routing protocols, by designing hybrid multicast schemes [11][12]. For example, the hybrid approach of geographic multicasting and topology-based (i.e., tree/mesh-based) multicasting [12] is used to find a good tradeoff between state information storage overhead, communication and computation overhead. Similarly, [11] proposes the hybrid solution of geographic multicast routing and source multicast routing. However, this recent research fails to find the near-optimal multicast routing topology, and does not consider the challenges posed by real deployments (e.g., deployments with obstructions, such as holes).

III. System Model and Problem Formulation

We consider a static wireless sensor network consisting of n nodes, denoted by $V = \{v_1, v_2, ..., v_n\}$, uniformly distributed in an area with obstructions, such as holes. There is one sink node denoted by S that collects data from other nodes. A subset of nodes $G \subseteq V$ form a multicast group rooted at S. We assume that each node knows its location and that the locations of multicast members are known to the sink node S. We also assume that existing multicast group management techniques (e.g., *join group*, *leave group*, etc.) are available. These techniques have been studied extensively in the literature. Consequently, in this paper, we focus exclusively on the energy efficient and robust multicast packet transmissions to the nodes in the multicast group.

The problem we address in this paper is two-fold. First, we aim to identify the near-optimal set of routing paths (i.e., near-optimal in the sense of minimizing the sum of path lengths) from the multicast root node to multicast members. Second, given the requirements imposed by large scale, real WSN deployments in complex environments, we aim to limit the packet header size (e.g., make it a system parameter), reduce the state information maintained by forwarding nodes, reduce the computational overhead in forwarding nodes, and achieve reliable packet delivery rate.

Inspired by the *Capacitated Concentrator Location Problem (CCLP)* [13][14], we aim to minimize the total sum of path lengths from the multicast root node node S to each multicast group member, by optimally selecting the locations of *facility* nodes f_i, as depicted in Figure 1 (a *facility* node– a term adopted from the operations research–can be thought of as the leader node in the hierarchical multicast routing).

From here on, we will use the term ***source*** node to represent the multicast root node, the term ***facility*** node to represent nodes aiding in the multicast routing from the root node to multicast group members, and the term ***member*** node to represent multicast group nodes.

For mathematically formulating our problem, we index the possible locations of facility nodes by i, and the possible locations of member nodes by j. We denote the member of

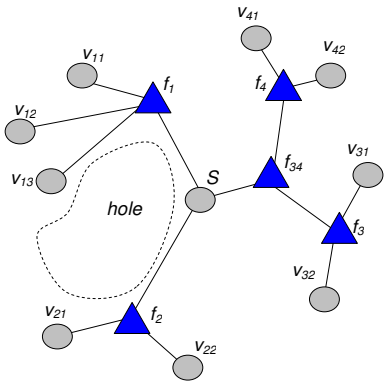


Figure 1. A multicast root node S is connected to facility nodes f_1, f_2, f_3, f_4 and f_{34}. Each facility node is connected to its member nodes v_{ij}, all members of the multicast group.

a facility node f_i by v_{ij} and the Euclidian distance between two nodes v_i and v_j by $d(v_i, v_j)$. Our problem can then be formulated as a mixed integer program:

$$\text{minimize} \quad \sum_i \sum_j d(f_i, v_{ij}) x_{ij} + \sum_i d(f_i, s) y_i \quad (1)$$

$$\text{subject to:} \quad \sum_i x_{ij} = 1 \text{ for all } j, \quad (2)$$

$$\sum_j x_{ij} \leq s_i y_i \text{ for all } i, \quad (3)$$

$$x_{ij} = \{0, 1\} \text{ for all } i \text{ and } j, \quad (4)$$

$$y_i = \{0, 1\} \text{ for all } i \quad (5)$$

where x_{ij} and y_i are the indicator variables (i.e., $y_i = 1$ if a facility node is available at location i, and $x_{ij} = 1$ if a member node at location j is connected to a facility node at location i); and s_i is the storage capacity of a facility node at location i, specifying the maximum number of members it can handle. The first constraint guarantees that a member node is connected to only one facility node. The second constraint specifies that no facility node can handle more than its storage capacity. The third and fourth constraints specify the integrality of x_{ij} and y_i.

Several challenges remain to be solved when CCLP is applied to large scale WSNs deployed in realistic environments. First, the *holes* with various shapes in real world deployments must be efficiently abstracted and taken into account when aiming for the optimal solution. The reason for including the holes in the protocol design is that they change the end-to-end communication cost (i.e., $d(f_i, v_{ij})$ and $d(f_i, S)$ in CCLP). The classical CCLP formulation assumes no *holes* in the target region. Thus, we need to reduce the multicast communication costs by: efficiently and proactively detecting the holes in the network; abstracting the hole information; and by recomputing the optimal solution. Second, CCLP assumes a high-speed and zero-cost communication medium between the source node and each facility; thus, CCLP typically ignores the communication cost for delivering a packet along the path between the source node and a each facility node. In our problem, however, the path from a source node to each facility node consists of multi hop wireless links, having similar characteristics to the links between facility nodes and their members. Consequently, by iteratively solving the CCLP problem (i.e., finding the new facility nodes with the existing facility nodes as new members) one can further optimize the multicast routing paths. Lastly, a reliable multicast packet delivery is typically required in real world WSN deployments. The reliability must be ensured with reduced recovery time and the small number of control packets. The design of RE^{2}MR addresses these problems in the sections that follow.

IV. Proposed Solution

In this section we provide an overview of RE^{2}MR, followed by the designs of its components.

A. Main Ideas

A key observation is that the topology obtained by solving the CCLP has the properties that satisfy our goals. In the topology obtained, the sum of Euclidean distances between a source node and members is near-optimal; the packet header size is limited to the capacity of a facility node (which can be set as a system parameter); the majority of computation happens at the source node, because the source node solves the centralized approximation algorithm for the CCLP and finds the set of facility nodes; facility nodes can be used for implementing reliable multicast packet delivery with small control packet overhead.

In order to obtain the near-optimal multicast routing path, it is important to detect/identify holes and provide the information on them (i.e., the size, shape, and location of the hole) to the source node (which solves the CCLP). Detecting the holes, however, requires high message overhead. Furthermore, the size of the packet must be large for precisely describing the information on the holes. One important observation we make is that only the holes affecting the optimal multicast routing paths must be detected. Based on theis observation, we propose a Trajectory-based Lightweight Hole Detection (TLHD) algorithm as part of our multicast protocol.

More aggressive energy savings can be achieved by finding a new set of facility nodes that serve existing facility nodes. However, to enable this multi-level facility system, a new message passing mechanism must be developed. Our Energy efficient Packet Forwarding (EPF) scheme provides an efficient way to deliver a packet to the facilities in multiple levels with reduced number of packet transmissions, by using the broadcast nature of wireless communications and careful packet header design.

Algorithm 1 RE^{2}MR Protocol

1: Init: $k \leftarrow 0$, $F_k \leftarrow G$, $r \leftarrow$ TRUE
2: **while** (packets to send) > 0 **do**
3: **if** $r =$ TRUE **then**
4: $F_{k+1} \leftarrow CCLP(F_k)$
5: **end if**
6: Forward a packet to each $f_i \in F_{k+1}$ using EPF
7: // Hole detection (TLHD) started
8: **if** Feedback received **then**
9: Update hole info.
10: $r \leftarrow$ TRUE
11: continue (i.e., goto Line 3) // Recompute solution
12: **else**
13: // Multi-level facility
14: **if** $k + 1 <$ FACILITY_LEVEL **then**
15: $k++$
16: $r \leftarrow$ TRUE
17: continue (i.e., goto Line 3)
18: **end if**
19: **end if**
20: $r \leftarrow$ FALSE
21: **end while**

B. Energy Efficient and Robust Multicast Routing (RE^{2}MR)

The proposed RE^{2}MR protocol, presented in Algorithm 1, consists of four main components: the CCLP solver, the TLHD algorithm, the EPF scheme, and Multi-level Facility computation. As shown, the $CCLP$ solver is used to compute the locations of the facility nodes in the first level (denoted by F_1). Subsequently, a packet is forwarded to each facility $f_i \in F_1$ (Line 2-6). The EPF algorithm is used to forward a packet to reduce the total communication costs and to enable the multi-level facility computation. During the packet transmission to each facility node, TLHD is used for detecting any holes that interfere with the path from the root node to the facility node. If a hole is found, a feedback packet is sent immediately to the root node. This feedback packet is used to efficiently estimate the size, shape, and location of the hole. If the feedback packet is received, the root node updates its database of the detected holes and recomputes the solution reflecting the newly discovered holes (Line 8-11). Otherwise, RE^{2}MR checks if the multi-level facility computation is enabled (Line 14). If it is enabled, RE^{2}MR recomputes a new solution F_2, in a similar way that it computed previous facilities in F_1. Otherwise, RE^{2}MR keeps forwarding the next packet to the facilities in the first level F_1 (Line 14-17). This recomputation process is repeated until RE^{2}MR finds the facility set F_k, where k equals FACILITY_LEVEL.

In the following subsections, we present the four main components of RE^{2}MR in detail.

Algorithm 2 Trajectory Based Hole Detection (TLHD)

1: **if** feedback_bit **then**
2: **if** $(| \perp (v_i, \overline{st})| > \tau)$ or (local_minimum) **then**
3: $Org_x \leftarrow x_i$ $Org_y \leftarrow y_i$
4: Force_Face_Routing()
5: **end if**
6: **if** $|x_i - x_{i-1}| > I$ **then**
7: find index idx_i corresponding to $| \perp (v_i, \overline{st})|$
8: **if** $x_i > x_{i-1}$ **then**
9: encode idx_i
10: **else**
11: encode $-idx_i$
12: **end if**
13: **end if**
14: **else**
15: **if** $(| \perp (v_i, \overline{st})| > \tau)$ or (local_minimum) **then**
16: detect_bit $\leftarrow 1$
17: **end if**
18: Forward()
19: **end if**

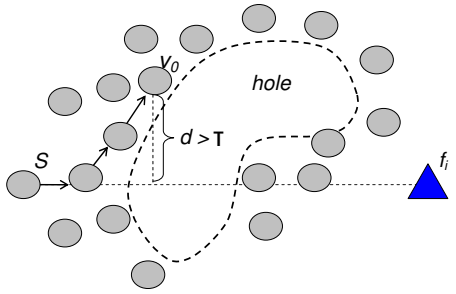


Figure 2. An illustration of hole detection. A packet, routed around the hole, measures the distance d to the line connecting the source S with facility node f_i. If the distance d is greater than a user defined threshold τ, a hole is detected.

C. Trajectory-based Lightweight Hole Detection (TLHD)

The TLHD algorithm, presented in Algorithm 2, consists of the hole detection phase and hole identification phase.

The hole detection phase is implemented as part of multicast packet transmission; thus, this phase does not require additional packet transmissions. Figure 2 illustrates the hole detection phase. As shown, if a forwarding node v_i, finds that it is in local minimum, or that the perpendicular distance to the line $\overline{Sf_i}$ is greater than a given threshold, then node v_i sets the *hole detection bit* in the packet header and forwards the packet (Lines 15-17). When the packet reaches facility node f_i, the facility node f_i checks the hole detection bit. If this bit is set, then the facility node f_i starts to execute the second phase, the hole identification phase.

In the hole identification phase, the information about the detected hole (i.e., the location, size, and shape of the hole) is obtained and concisely represented. Figure 3 illustrates

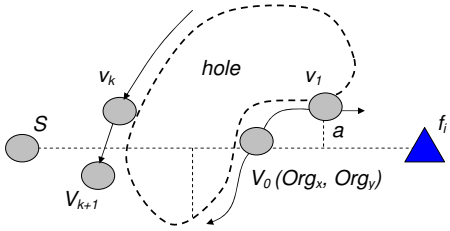


Figure 3. Illustration of hole identification: starting from the origin node v_0, a feedback packet traverses in clockwise direction (a copy of a feedback packet to counter clockwise direction) and records the perpendicular distance to a line $\overline{Sf_i}$ connecting a source S and a facility node f_i, until it crosses the line $\overline{Sf_i}$.

the hole identification phase. The facility node f_i that has received a packet with the *hole detection bit* set, starts this phase by sending a feedback packet to the source S. If the feedback packet reaches a forwarding node (v_0 in Figure 3) either in local minimum, or having a perpendicular distance to the line $\overline{Sf_i}$ greater than a given threshold (Algorithm 2: Line 2), the forwarding node v_0 encodes its location, (Org_x, Org_y) in the packet and, using face-routing, forwards the packet in the clockwise direction (Algorithm 2: Line 3-4). We call this forwarding node, an *origin node*. Additionally, the origin node sends a copy of the feedback packet in counter-clockwise direction. These two feedback packets will be routed in opposite directions around the hole, collect the information about the hole (described below), and meet at one boundary node of the hole. The collected information is combined into a one feedback packet, and transmitted back to the source node.

More specifically, while the nodes along the boundary of the hole route the packet, they measure the distance between them and the line $\overline{Sf_i}$. For example, consider Figure 3, where a packet from node v_0 reached the next boundary node along the face, node v_1 with location (x_1, y_1). Node v_1 computes the projected distance to previous node v_0 (i.e., $|x_i - x_0|$) and checks if this projected distance is greater than I. If it is greater, node v_1 calculates the perpendicular distance to line $\overline{Sf_i}$, represented by a in Figure 3. The representation of this distance is further abstracted as a simple index in a table, in which each entry of the table represents a range of distances. (Algorithm 2: Line 6-7). The matching index is then encoded in the feedback packet. In order to differentiate the packet forwarding directions (i.e, either towards the origin node or not), we use a negative representation of the index in the packet when the packet travels towards the source node (Algorithm 2: Line 8-12). The feedback packet is kept forwarded to the next boundary node along the hole, until it crosses line $\overline{Sf_i}$. For example in Figure 3, when the packet is forwarded from node v_k to node v_{k+1}, the line $\overline{v_k v_{k+1}}$ crosses the line $\overline{f_i S}$. And then, node v_{k+1} stops forwarding the feedback packet and waits for the feedback packet coming from the opposite direction.

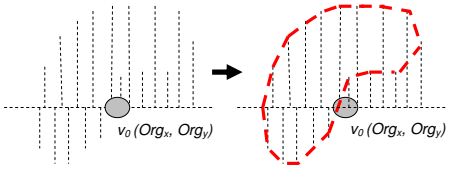


Figure 4. Illustration of hole reconstruction: the graph on the left side shows the set perpendicular distances to $\overline{st}$ starting from a origin node, and the graph on the right side shows reconstructed hole by sequentially connecting the end point of the perpendicular lines starting from the origin node.

If that packet arrives, node v_{k+1} combines the collected data and sends the packet to the source S.

A special case occurs when a facility node is located in a hole. The TLHD algorithm efficiently handles this situation. Assume that a packet is sent from the source S to a facility node f_i, and that this facility node is inside a hole. This packet would traverse along the hole, cross the line defined by S and f_i at a point p (for which $d(S, f_i) < d(S, p)$), and then be received by a node, say v'. Finally, node v' sets its feedback_bit, becomes an origin node, and initiates the hole identification phase to find a new facility node that is not inside a hole.

Upon receiving the feedback packet, the source node S uses the information on the detected holes for recomputing the optimal path to each member. Source node S firsts reconstructs a hole by using the data in the feedback packet. Figure 4 illustrates this process. The hole is represented as an origin (org_x, org_y) and a set of perpendicular distances to the line $\overline{f_i s}$ as shown in Figure 4. Consequently, source node S is able to represent the hole as a polygon by sequentially connecting all the end points of the perpendicular lines starting from the origin. This polygon representation of a hole is used to recompute the shortest path between a facility node and its members and between the facility node and the source node S. To compute such shortest path, RE^{2}MR exploits the *Visibility Graph*, a well known mechanism to compute the shortest path in the presence of polygonal obstacles [15].

D. Energy-efficient Packet Forwarding (EPF) and Multi-Level Facility Computation

There are two types of nodes in our protocol, namely facility nodes and non-facility nodes. In order to save energy by reducing the number of packets transmitted, RE^{2}MR uses different packet forwarding schemes for different node types. A facility node forwards a data packet to multiple destinations, either to its members, or to the facility nodes in the lower level. A naive forwarding scheme for a facility node is to use multiple unicasts to each destination. However, this method not only incurs high energy consumption but also causes unbalanced energy distribution. To solve this problem, our EPF scheme exploits the broadcast nature of wireless transmission and a careful packet header design.

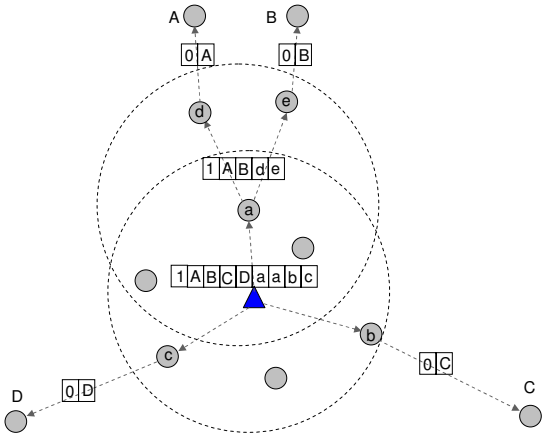


Figure 5. An illustration of packet forwarding by a facility node: the best next hops for destinations A, B, C and D are a, a, b and c respectively. In particular, node a determines the best next hops for the two destinations A and B.

Specifically, a facility node first computes the best neighbor for each destination (the best neighbor refers to the closest node to a given destination). The facility node then puts the locations of the destinations in the header and sequentially inserts the corresponding node id of the best neighbor for each destination. The first bit of the header is set to 1 so that a receiver treats this packet differently from a simple forwarding.

Upon receiving a packet, a node checks the first bit of the header. If this bit is set, the node checks if its node id matches any node ids in the header. In case of no-match, it just forwards this packet. If its node id matches the i-th node id in the header, it sets the destination location as the i-th location in the header. Figure 5 illustrates a packet forwarding scenario and the packet header. In this figure, a facility node has four member nodes with the locations A, B, C, and D respectively. The facility node first computes the best neighbors: b for C, c for D, and a for both A and B. It then sets the first bit of header to 1, puts the locations of the destinations, and inserts corresponding ids of best neighbors. Consequently, we have the header $1, A, B, C, D, a, a, b, c$. The facility node broadcasts this packet, so that all of its neighbors receive this packet. Upon receiving this packet, a node b finds that it has to forward this packet to C by looking at the header. Similarly, c forwards the packet to D. However, a node a finds that it has two destinations, A and B. Node a then applies the same logic to split the packet.

One other type of a node in our protocol is the non-facility node. Unless the first bit of a packet header is 1, a non-facility node simply forwards the packet to a destination using simple geographic forwarding, or face-routing for escaping from the local minimum.

As shown in Lines 14-17 of Algorithm 1, EPF also

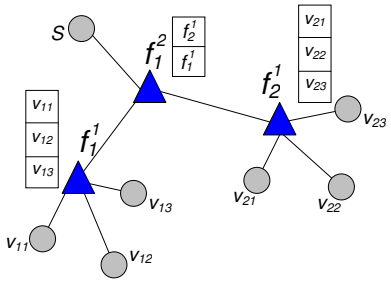


Figure 6. An illustration of multi-level facility nodes: previous two paths connecting S to f_1^1 and f_2^1 are further optimized by introducing a facility node f_1^2 in one higher level.

provides an efficient message passing mechanism for multi-level facility system. We illustrate the concept in Figure 6 where f_i^1 represents a facility node in the first level, f_i^2 means a facility node in the second level, and so on. Recall that in the single level facility system, a source node S sends a multicast packet only to the facility nodes in the first level. In the multi level facility system, a source node S first sends a multicast packet to the facility nodes f_i^1 in the first level; then each facility node f_i^1 saves the locations of its members (which are encoded in the header of the received packet). A source node S then sends the second multicast packet to the facility nodes f_i^2 in the second level. Similarly, each second-level facility node f_i^2 saves the locations of its members (in this case, the members are the facility nodes in the first level), and forwards the packet to its members. Upon receiving the second multicast packet from the facility node in the second level, the facility node in the first level forwards the packet to its members, using locally stored locations. This process is repeated for higher facility levels. Consequently, a source node sends a packet to the facility nodes at the highest level without encoding the locations in the packet header.

Note that adding more facility levels permits higher energy savings. However, in order to use higher facility levels, more nodes need to maintain state information (i.e., the locations of members). RE^{2}MR allows the user of a WSN to make the tradeoff decisions between more aggressive energy savings and smaller state information.

E. Reliable Packet Delivery

To improve the packet delivery ratio, we employ a slightly modified version of NACK based retransmission scheme, which is particularly useful for RE^{2}MR. Specifically, the recovery time from a packet loss is reduced, since the retransmission request is directly sent to the facility node, not to the source node. Additionally, since the facility nodes handle packet retransmission requests, we eliminate the bottleneck that the source node would have been, had it handled all retransmission requests.

Algorithm 3 presents the Reliable Packet Delivery (RPD)

Algorithm 3 RPD: Code for facility node f

1: Init: $\mathcal{B} \leftarrow \emptyset$
2: On received packet p_i:
3: $\mathcal{B} \leftarrow \mathcal{B} \cup \{p_i\}$
4: **if** p_i in sequence **then**
5: Forward to members
6: Start timer T_i
7: **else**
8: Send NACK$_j$ to the root (j:the index of missed packet)
9: **end if**
10: On timer T_i fired:
11: **if** NACK not received **then**
12: $\mathcal{B} \leftarrow \mathcal{B} \setminus \{p_i\}$
13: **else**
14: Retransmit p_i
15: Reset timer T_i
16: **end if**

scheme implemented by facility nodes. As shown, upon receiving a packet p_i from a source node (or from a facility node in one level higher than itself, when multi-level facility are used), a facility node f, checks if the packet p_i is in sequence. If it is in sequence, f simply forwards p_i to its members and sets a timer T_i, specifying the waiting time for NACK packets from its members (Line 4-5). If the facility node identifies a lost packet p_j, then it sends NACK$_j$ to the root node (Line 7-9). When the timer T_i fires, f examines if any NACK packets have been received from its members. If no NACK packets were received, f deletes the packet p_i from buffer $\mathcal{B}$ (Line 11-12). If a NACK$_i$ was received, f retransmits the packet p_i and resets the timer T_i (Line 13-15).

V. Theoretical Analysis

In this section we present an analysis of RE^{2}MR and two state of art multicast routing protocols, namely RSGM [8] and MRBIN [12]. For our analysis, we consider a two dimensional network with a grid topology of varying size, $n \times n$, where n is an even positive number. The inter-node distance is a unit distance. Member nodes are located along the four edges of the network (i.e., $4n$ members are positioned in the network of size $n \times n$). The total number of branch nodes, denoted by N_b, is an indicator that represents the total amount of state information maintained in the network. The following results show the total number of branches when there are N members. The sum of path lengths is indicative of how much energy is consumed for each multicast packet transmission. The following analysis estimates the sum of path lengths as a function of N members. The proofs are in the Appendix.

Theorem 1: RE^{2}MR has the lowest N_b, when compared with RSGM and MRBIN.

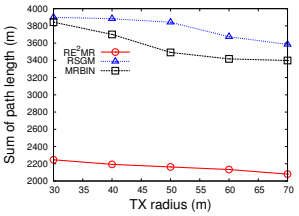


Figure 7. Impact of node density on sum of path lengths.

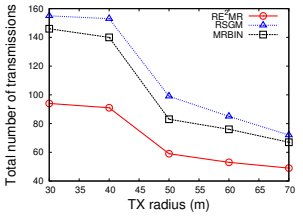


Figure 8. Impact of node density on total number of packets.

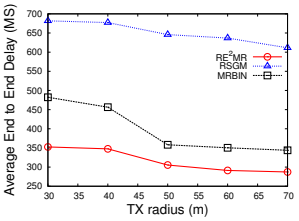


Figure 9. Impact of node density on average end-to-end delay.

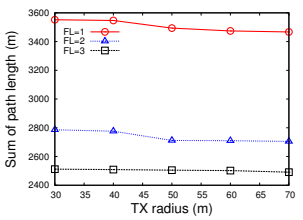


Figure 10. Impact of facility level on sum of path lengths.

Theorem 2: RE^{2}MR has the shortest sum of path lengths, when compared with RSGM and MRBIN.

VI. Performance Evaluation

We implemented RE^{2}MR in nesC for the TinyOS operating system. The protocol (implemented in 4,916 lines of code) occupies 7,289B of RAM, and 25,466B of program memory. We adopted GPSR [16] for the underlying geographic routing protocol. We compared the performance of RE^{2}MR with two recent, state of art multicast protocols for WSN: one hierarchical geographic multicast–RSGM [8], and one hybrid multicast–MRBIN [12]. Due to the relatively large scale network deployment we need for our evaluation, we performed simulations using TOSSIM. For our simulations, we deployed 400 nodes in a 20×20 grid, with an inter-node distance of 20m. The radio range of a node was between 30m and 70m. The metrics we used for our performance evaluation are: the total sum of path lengths (PL), the total number of packets transmitted (PC), the average end-to-end delay ($E2E$) and the packet delivery ratio (PDR). We vary the following parameters: node density (ND), facility level (FL), and hole size (HS). Each experimental point represents the mean of five runs.

A. Impact of Node Density

We expect that node density (ND) affects the performance of RE^{2}MR, RSGM and MRBIN, because these protocols are based on geographic routing that is known to be sensitive to node density. We measured PL, PC and $E2E$ by varying ND from 30 to 70. For this experiment, we fixed FL=1, FC=3, and NM=4%. Figure 7 depicts the results for PL. One can observe that RE^{2}MR yields shorter path lengths, by as much as 57%, when compared with both RSGM and MRBIN. As shown, when ND increases,

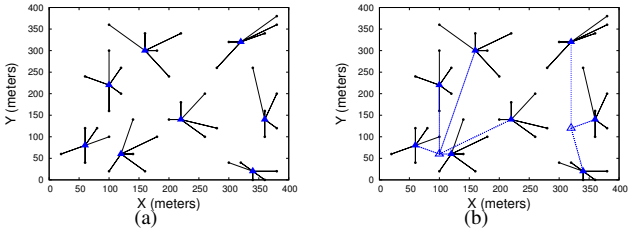


Figure 11. Example of RE^{2}MR topologies for (a) a single level facility and (b) two level facilities.

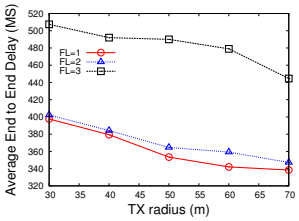


Figure 12. Impact of facility level on average end-to-end delay.

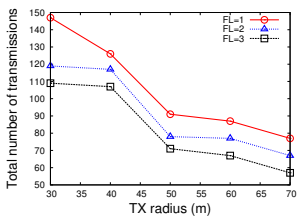


Figure 13. Impact of facility level on total number of communications.

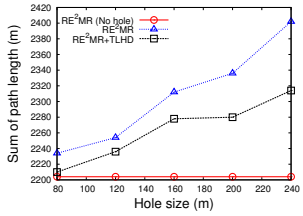


Figure 14. Impact of hole size on sum of path lengths.

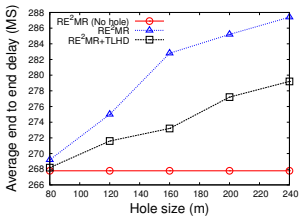


Figure 15. Impact of hole size on average end-to-end delay.

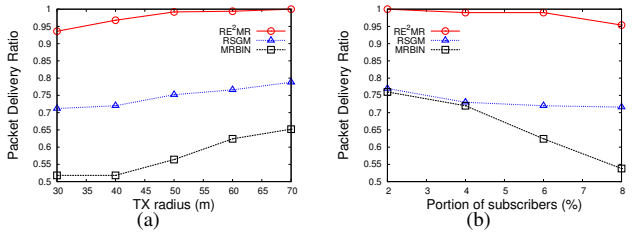


Figure 16. Reliability measurements: (a) as a function ND; and (b) as a function of NM.

PL for all the protocols decreases slightly. The explanation for this is that at higher node densities, geographic routing protocols are able to identify routing paths more closer to the Euclidian distance between a source and a destination, and, hence, shorter. Figure 8 shows PC as a function of ND. As expected, for larger communication ranges, the total number of packets exchanged decreases. One can observe that the PC for RE^{2}MR is the lowest for all the ND values. Figure 9 depicts the end-to-end delay $E2E$ for the three protocols as a function of ND. The results indicate that protocols with lower PL exhibit a lower end-to-end delay. As shown, RE^{2}MR has an average end-to-end delay shorter by up to 8% when compared with MRBIN, and by up to 50% when compared with RSGM.

B. Impact of Level of Facilities

In this subsection, we investigate how the facility level (FL) affects the performance of our protocol. Specifically, we measured PL, PC and $E2E$ by varying FL in different ND settings. For this experiment, we fixed NM=6% and FC=3 and did not consider the holes. Example topologies for RE^{2}MR with single level and two level facilities are depicted in Figure 11.

Figure 10 shows PL for different FL values. Similar with the results presented in Figure 7, PL slightly decreases as ND increases, for all FL values. As expected, higher FL results in shorter PL. The reason is that the path lengths from a source node to facility nodes are reduced when higher level facility nodes are used. Figure 13 depicts PC as a function of ND for different FL. Note that PC becomes smaller for longer communication ranges. We also observe that PC for higher FL was lower than for lower FL. The reason is that higher FL essentially aggregates more paths. However, as Figure 12 depicts, $E2E$ for higher FL is actually higher than for lower FL. Although the total sum of path lengths is reduced by aggregating more existing paths using facility nodes in higher levels, the direct path to a facility node in lower level was no longer used, causing higher end-to-end delay for higher facility level.

C. Impact of Holes

To investigate the impact of a hole, we created a hole in the middle of our network. Specifically, the hole is a square shape with the length of a side varying from 80m to 240m. In this experiment we fixed ND=60, FC=3, FL=1 and we uniformly deployed 12 members along the upper and right sides of our network, allowing space for the hole. We measured PL and $E2E$ by varying the size of the hole.

Figures 14 and 15 depict the results. When a hole is present, the performance in terms of PL and $E2E$ degrades, when compared with the scenario when a hole is not present. As we increase the size of the hole, the performance degradation increases. The reason for this is that a hole affects the routing cost between a facility node to the source node, and to members, by making a packet travel along the face of the hole. Evaluation results demonstrate how our proposed TLHD algorithm improves the performance by relocating the facilities. The TLHD algorithm allows the source node to recalculate the locations of the facility nodes. As we increase the size of the hole, PL and $E2E$ for RE^{2}MR with TLHD also increase. The impact of the hole, however, is mitigated by the new set of facilities.

D. Reliability

In this section, we investigate the reliability of RE^{2}MR by measuring the packet delivery ratio (PDR). We compared the

PDR of RE^{2}MR to that of RSGM and MRBIN for different ND and NM settings. We fixed NM=6%, FC=3, FL=1 and measured PDR by varying ND from 30m to 70m. The source node sent 100 packets, at a rate of 2 packets per second. Each member node computed its own PDR. The reported PDR was then calculated as the average PDR of all members.

Figure 16(a) depicts our results. As shown, as we increase ND, the PDR values of all three protocols increase. The reason is that an increased transmission range reduces the number of packet transmissions, thus decreasing the probability of packet collisions. Additionally, a higher node density increases the probability of packet delivery. Note that RE^{2}MR's PDR approaches almost 100%, greatly outperforming MRBIN and RSGM. The low PDR of state of art multicast protocols suggests that a packet recovery mechanism must be employed for reliability. An additional observation is that MRBIN's PDR is worse than that of RSGM. The number of branch nodes in RSGM depends on the size of the cell. In our setting, RSGM has more branch nodes, when compared with MRBIN; thus MRBIN shows relatively better performance in terms of PL, PC and $E2E$. However, the smaller number of branch nodes means the higher chance of packet loss, because a single packet carries data to more members. This also explains why MRBIN's PDR rate increase is higher than that of RSGM.

Next, we fixed ND=60, FC=3 and FL to 1, and measured PDR for RE^{2}MR, RSGM, and MRBIN by varying NM from 2% to 8%. Figure 16(b) shows the result. As we increase NM, PDR for all protocols decreases. A simple explanation for this is that higher traffic increases the chance of collisions and interference. Similar to the result for different ND setting, RE^{2}MR shows the best performance, when compared with RSGM and MRBIN. Similar to the result shown in Figure 16(a), MRBIN's decrease rate in PDR for increasing NM was higher than that of RSGM, because MRBIN has the smaller number of branch nodes.

VII. Conclusions

This paper presents RE^{2}MR, an energy efficient and robust geographic multicast routing protocol for large scale WSN deployments in realistic environments (e.g., characterized by obstructions, such as holes). We demonstrate RE^{2}MR's practicality through a TinyOS implementation for motes and its effectiveness through an extensive performance evaluation in TOSSIM, which includes comparisons with recent, state of art multicast routing protocols.

References

[1] J. G. Jetcheva and D. B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," in *Proc. of ACM MOBIHOC*, 2001.

[2] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-demand multicast routing protocol," in *Proc. of IEEE WCNC*, 1999.

[3] M. Mauve, H. Fussler, J. Widmer, and T. Lang, "Position-based multicast routing for mobile ad-hoc networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 53–55, 2003.

[4] J. Sanchez, P. Ruiz, and I. Stojmnenovic, "Gmr: Geographic multicast routing for wireless sensor networks," in *Proc. of IEEE SECON*, 2006.

[5] S. Wu and K. Candan, "GMP: Distributed geographic multicast routing in wireless sensor networks," in *Proc. of IEEE ICDCS*, 2006.

[6] D. Koutsonikolas, S. M. Das, Y. C. Hu, and I. Stojmenovic, "Hierarchical geographic multicast routing for wireless sensor networks," *Wireless Network*, vol. 16, no. 2, pp. 449–466, 2010.

[7] S. M. Das, H. Pucha, and Y. C. Hu, "Distributed hashing for scalable multicast in wireless ad hoc networks," *IEEE Transactions on Parallel Distributed System*, vol. 19, no. 3, pp. 347–362, 2008.

[8] X. Xiang, X. Wang, and Y. Yang, "Stateless multicasting in mobile ad hoc networks," *IEEE Trans on Computers*, vol. 59, 2010.

[9] X. Zhang and L. Jacob, "Multicast zone routing protocol in mobile ad hoc wireless networks," in *Proc. of IEEE LCN*, 2003.

[10] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks," *Cluster Computing*, vol. 1, no. 2, pp. 187–196, 1998.

[11] S. Song, D. Kim, and B.-Y. Choi, "AGSMR: Adaptive geo-source multicast routing for wireless sensor networks," in *Proc. of WASA*, 2009.

[12] S. Song, B.-Y. Choi, and D. Kim, "MR.BIN: Multicast routing with branch information nodes for wireless sensor networks," in *Proc. of IEEE ICCCN*, 2010.

[13] R. Sridharan, "The capacitated plant location problem," *European Journal of Operational Research*, vol. 87, no. 2, pp. 203–213, 1995.

[14] H. Pirkul, "Efficient algorithms for the capacitated concentrator location problem," *Comput. Oper. Res.*, vol. 14, no. 3, pp. 197–208, 1987.

[15] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan, "Linear time algorithms for visibility and shortest path problems inside simple polygons," in *Symposium on Computational Geometry*, 1986.

[16] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. of ACM MOBICOM*, 2000.

VIII. Appendix

A. Proof of Theorem 1

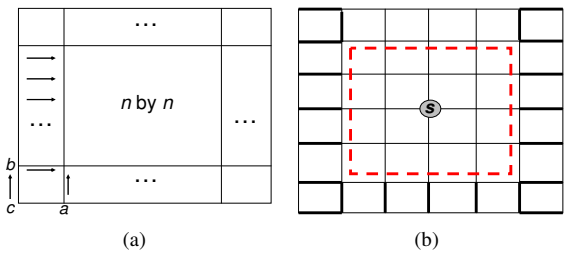
Lemma 1 and Lemma 2 are first proved to prove Theorem 1.

Lemma 1: For MRBIN, maximum $N_b = N - 6$.

Proof: We prove this by induction on the length of side n of our network.

Base step ($n = 2$, i.e., 2×2 network): it is trivial to see that $N_b = 2$ for a 2×2 network, where the number of members is 8.

Inductive Step: assume that for $n = k$, $N_b = 4k - 6$. Now consider Figure 17(a). A path from each member at the corner of $(n+1) \times (n+1)$ network meets with the path from



(a) (b)

Figure 17. Illustration of inductive step for: a) Lemma 1 and b) Lemma 3.

one of its adjacent members, making four branches. Except for these members, all the other members send packets to the adjacent member that is located on the edge of $n \times n$ network as shown in Figure 17(a). Therefore, the total number of branches for the $(n+1) \times (n+1)$ network is given by $N_b = (4k-6)+4 = 4(k+1)-6$. □

Lemma 2: For RSGM, maximum $N_b = \frac{N}{2} - 4$.

Proof: In RSGM, a branch node is a leader in a cell. Thus, we need to consider the total number of cells that have members in them. If the cell size is C_s, in a $n \times n$ network, the total number of cells is $\frac{n^2}{C_s}$. Next, we need to consider the cells that contain members (i.e., the cells that are adjacent to the four sides of the network). The number of cells that do not contain members can be computed as $\frac{(n-2\sqrt{C_s})^2}{C_s}$. Thus, the total number of branches is $N_b = \frac{n^2}{C_s} - \frac{(n-2\sqrt{C_s})^2}{C_s}$. Next, by replacing n by $\frac{N}{4}$ and solving the equation for N_b, we obtain $N_b = \frac{N}{\sqrt{C_s}} - 4$, where $4 \leq C_s < \frac{N^2}{16}$. Therefore, the maximum $N_b = \frac{N}{2} - 4$. □

Theorem 3: RE^{2}MR has the lowest N_b, when compared with RSGM and MRBIN.

Proof: In RE^{2}MR, facility nodes are the only branch nodes. Thus, we count the maximum number of facility nodes. Let F_c be the facility capacity. Since members are uniformly located along the four sides of the network, $\lceil \frac{N}{F_c} \rceil$ facilities are selected such that members around the corners of the network are first covered, thereby $\lceil \frac{N}{F_c} \rceil \geq 4$. The maximum number of facilities is N when the facility capacity $F_c = 1$. However, for fair comparison, we must determine the value for F_c. Note that for RSGM, L_b is minimum when C_s=4, which yields that the capacity of a leader, specifically the leader of a cell located at the corner of network, is at most 5. For MRBIN, the capacity is at most 4 due to the grid topology. Thus, we choose 5 for F_c and get $N_b = \lceil \frac{N}{5} \rceil$. By Lemma 1 and Lemma 2, MRBIN $(N-6)$ > RSGM $(\frac{N}{2} - 4)$ > RE^{2}MR $(\lceil \frac{N}{5} \rceil)$. □

B. Proof of Theorem 2

Lemma 3 and Lemma 4 are proved first to prove Theorem 2.

Lemma 3: The sum of path lengths for MRBIN is $\frac{N^2}{16} + \frac{N}{2}$

Proof: We prove this by induction on the length of side n of n by n network.

Basis step ($n = 2$): the path length is trivially 8.

Inductive step: assume that the claim holds for $n = k$. Then, the path length of $k \times k$ network is $k^2 + 2k = k(k+2)$, since $N = 4k$. Now we consider the case with $n = k+2$, since n is an even positive number. Figure 17(b) depicts this case. Note that each member can reach the inner square, $k \times k$ network, in one hop. In particular, the members at the four corners reach the inner square by forming a branch with one of its two neighboring members. Thus, we get $4(k+2)$ additional path lengths, which yields that the total path length for $n = k+2$ is $k(k+2) + 4(k+2) = (k+2)(k+4)$. □

Lemma 4: The sum of path lengths for RSGM is $\frac{3}{2}(\frac{N^2}{16} - \frac{N}{2} + 8)$.

Proof: In order to compute the sum of path lengths, we consider a Cartesian coordinate system in which the member at the left bottom corner of the network is the origin. We first compute the path lengths from the source node S to each leader. The longest path is the one that connects the source node with the leader of the cell located at the corner of the network. The coordinates of this leader are $(\frac{\sqrt{C_s}}{2}, \frac{\sqrt{C_s}}{2})$, and the coordinates of S are $(\frac{N}{8}, \frac{N}{8})$. The Euclidean distance between them is $\sqrt{2}(\frac{N}{8} - \frac{\sqrt{C_s}}{2})$, and thus the actual path length is $2(\frac{N}{8} - \frac{\sqrt{C_s}}{2})$. The shortest path is of length $\frac{N}{8} - \frac{\sqrt{C_s}}{2}$. Thus, the median path length is $\frac{3}{2}(\frac{N}{8} - \frac{\sqrt{C_s}}{2})$. When taking into account the total number of leaders, the sum of lengths for the paths connecting a source to leaders is $\frac{3}{2}(\frac{N}{8} - \frac{\sqrt{C_s}}{2})(\frac{N}{\sqrt{C_s}} - 4)$. Now we compute the path lengths from a leader to members. The longest path connects a leader to its members located at the left bottom corner of its cell. This member has coordinates $(0,0)$. The Euclidean distance between them is $\frac{\sqrt{2C_s}}{2}$, and thus the actual distance is $\sqrt{C_s}$. The shortest length of such path is $\frac{\sqrt{C_s}}{2}$. Thus, the median length is $\frac{C_s + \sqrt{C_s}}{4}$. Since the total number of members is N, the sum of lengths of such paths is $(\frac{C_s+\sqrt{C_s}}{4})N$. Therefore, after combining the results for the two cases above, the total sum of path lengths becomes $\frac{3}{2}(\frac{N}{8} - \frac{\sqrt{C_s}}{2})(\frac{N}{\sqrt{C_s}} - 4) + (\frac{C_s+\sqrt{C_s}}{4})N$. When considering C_s=4, for maintaining a minimum number of branch nodes for MRBIN, we obtain $\frac{3}{2}(\frac{N^2}{16} - \frac{N}{2} + 8)$. □

Theorem 4: RE^{2}MR has the shortest sum of path lengths, when compared with RSGM and MRBIN.

Proof: The maximum distance between a source node S and the facility node is $2(\frac{N}{8} - 2)$, when the facility node is located at the point closest to the member positioned at the corner of network. By Lemma 2, we choose F_c=6. Thus, the sum of path lengths connecting S and facilities is at most $2(\frac{N}{8} - 2)\frac{N}{6}$. The maximum path length between the facility node and its member is thus $\sqrt{10}$. Since there are N members, the sum of path lengths connecting a facility node to each member is at most $\sqrt{10}N$. Thus the total sum of path lengths is at most $2(\frac{N}{8} - 2)\frac{N}{6} + \sqrt{10}N$. □