A Dynamic Localized Minimum-Energy Agent Tree-Based Data Dissemination Scheme for Wireless Sensor Networks

Ping Ding, JoAnne Holliday
Computer Engineering Department, Santa Clara University
{pding, jholliday}@scu.edu

Abstract: The problem of efficiently disseminating data from a mobile stimulus (source) to multiple mobile users (sinks) in a large-scale sensor network has been challenging. We address this problem by proposing a minimum-energy tree-based data dissemination scheme, *Dynamic Localized Minimum-Energy Agent Tree-Based Scheme* (DLATS). We exploit the fact that sensor nodes are stationary and location-aware. In DLATS, each sensor node finds its *Relative Neighborhood Graph* (RNG) neighbors and a RNG is generated over the whole network. Then, the source broadcasts its position information to all the other nodes using our localized minimum-energy broadcast protocol, *Improved RNG Broadcast Oriented Protocol* (IRBOP). A dynamic agent tree is generated between each source and multiple sinks using our *Shortest Direct Parent First* (SDPF) where the sinks become the agents, e.g., the leaves of the agent tree. Finally, each source uses IRBOP for multicasting the stimulus data to the users over the agent tree. We evaluate the performance of DLATS through simulations. Results show DLATS outperforms previously proposed protocols for data dissemination in large-scale sensor networks.

Keywords: Dynamic Localized, Minimum-Energy, Data Dissemination, Sensor Networks

1 Introduction and Related Work

Because of the limited power supplies, there are several research issues in such sensor networks: 1) How to efficiently detect stimulus or users; for example, [1] summarizes scalable location services and compares their performances. 2) How the sinks collect data from the sources. There are several research issues in this topic: a) How to locate the sources to the sinks or how to locate the sinks to the sources; b) How to report each source's data to multiple sinks; that is, how to do energy-efficient multicast between them; c) How to efficiently reconfigure or regenerate the links between each source and the multiple sinks when either the source or the sinks are reassigned; that is, how to maintain the energy-efficient multicast between them. In this paper, we focus on these three issues. To the best of our knowledge, there has not been any published work that efficiently solves all three problems simultaneously.

To minimize the total energy consumption in broadcasting but still enable a message originating from a source node to reach all the other nodes, the source node can adjust its transmission power. This is called the minimum energy broadcasting problem [2]. Minimum energy broadcasting problem can be solved by topology control protocols which aim to adjust transmission power while preserving strong connectivity of the network. *Relative neighborhood graph* (RNG) has been used in topology control protocol [3]. Cartigny et al. [2] propose a localized minimum-energy broadcasting protocol, RNG broadcast oriented protocol (RBOP). RBOP broadcasts to one-hop neighbors in a RNG. In this paper, we argue that a less greedy, two-hop broadcast approach where each node is less greedy because it considers the energy expenditure of its neighbors when deciding its own transmit radius. And, we show it save energy.

Li et al. [4] propose a location system to estimate the target position as well as its motion, which takes advantage of the independently calculated time-difference-of-arrival of successive pulses from the targets. In [5], authors give the directed diffusion paradigm to achieve energy saving by selecting paths between the sources and the sinks. In [6], authors propose the two-tier data dissemination (TTDD) scheme where each source needs to generate a grid to tell its position to all other sensors. Sensor nodes located at the grid points work as agents. Then, each sink communicates with the source through the agents. Both of the two algorithms provide multicasting between one source and multiple sinks, however, they do not provide efficient multicasting and are not efficient when the sources and the sinks are mobile. In [7], authors build a dynamic proxy tree-based data dissemination scheme where it realizes minimum-energy multicasting between each source and the sinks based on a centralized algorithm.

Lou et al. [8] perform dominant pruning and partial dominant pruning to reduce broadcast redundancy. Cagalj et al. [9] creates a heuristic, Embedded Wireless Multicast Advantage algorithm which realizes a minimum-energy broadcast and takes the advantage of relaying nodes. Wu et al. [10] study how to realize efficient broadcasting using a small set of forwarding nodes. All of the previous algorithms do not guarantee the total consumed energy is within a constant factor of the optimum. Wan et al. [11] have proved that minimum-energy broadcasting based on Euclidean Minimum Spanning Tree (MST) or MST-based graph consumes energy within a constant factor of the optimum. Since MST cannot be constructed in a localized manner, a localized approximation structure of MST, RNG, has been suggested for broadcasting.

The contributions of the paper are: 1) the localized data dissemination scheme from each mobile stimulus to multiple mobile users, *Dynamic Localized Minimum-Energy Agent Tree-Based Scheme*, DLATS; 2) efficient broadcasting via two-hop neighbors, *Improved RBOP*; 3) building an agent tree between each mobile stimulus to multiple mobile users where the energy paths from the agents to the root are the minimum, *Shortest Direct Parent First*, SDPF; 4) fast and low-cost reconfiguration and regeneration of the agent tree. The paper is organized as follows: we propose DLATS in Section 2. We present our performance studies in section 3 and conclude in section 4.

2. Dynamic Localized Minimum-Energy Agent Tree-Based Scheme

2.1 Network Model

We assume that a sensor network has the following properties:

1) Nodes are dispersed in a 2-dimensional space and cannot be recharged after deployment; 2) Nodes are stationary once deployed. Both the stimuli and the users are mobile; 3) Nodes transmit at the same fixed power levels which are dependent on the transmission distance; 4) Nodes base decisions on local information only; 5) Nodes are location-aware, which can be defined using GPS, signal strength, or direction; 6) The energy consumption among nodes is not constrained to be uniform.

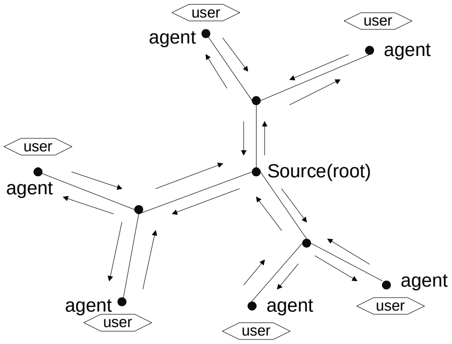

Figure 1 A dynamic agent tree

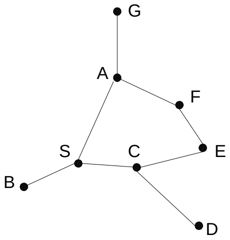

Figure 2 A RNG

2.2 Design Objectives

1) The network connectivity is preserved with the use of minimum possible power. This is the most important objective of minimum-energy broadcasting.
2) DLATS is distributed. This is because the network may be composed of thousands of nodes. It is expensive to know all other node's information. To run DLATS, a node only needs to know its local information.
3) DLATS is not affected by mobility. When either stimuli or users or both of them move, the minimum-energy broadcasting or multicasting can be guaranteed.
4) All links are bi-directional. For example, there are two nodes u and v in the network. If u can reach v then v can reach u.
5) DLATS expects a node's degree to be small but does not require the node's degree to be minimal. This is because: a) a small node degree may help to mitigate the hidden nodes problem; b) a small node degree can not guarantee that the graph approximates the optimal broadcast structure in terms of the total energy consumption [12].

2.3 DLATS Framework

Due to the dynamic characteristics of the sources and the sinks, it is difficult to maintain a tree that directly connects a source and multiple sinks that are interested in the source, or disseminate data from the source to the sinks. To realize minimum-energy multicasting, we generate a *dynamic agent tree*, which is a subset of RNG, between a source and the sinks. Figure 1 shows a dynamic agent tree between the source and the sinks. The source is the root and the sinks are the leaves in the tree. To generate the tree, a source implements our minimum-energy broadcasting protocol, IRBOP, over the RNG graph. Sink nodes are agents of the mobile users and join the tree to obtain information on behalf of the users. After a sink joins the tree, it becomes an agent. Each node that wants to be part of the tree will find an optimal (minimum-energy) path to the root based on a heuristic algorithm, *Shortest Direct Parent First* (SDPF). After each node finds its optimal path, a dynamic agent tree is generated. The source implements minimum-energy broadcasting over the agent tree. Effectively, this is akin to multicasting to a set of nodes within the entire network. When the users move, the sink nodes change and the agent tree will be efficiently reconfigured based on a localized algorithm, *Inform Direct Parent Only* (IDPO). If the stimulus moves and a new source is assigned, the new source runs IRBOP over the same RNG to tell its position to other nodes and the dynamic agent tree can be easily regenerated.

2.4 DLATS

DLATS consists of five steps.

1) Finding neighbors: each node finds its RNG neighbors; 2) Minimum-energy broadcasting: each source node broadcasts its position information to all other nodes using a minimum-energy broadcast protocol, *Improved RNG broadcast oriented protocol*, IRBOP(A); 3) Dynamic agent tree generating: each agent finds its minimum-energy path to the source using *Shortest Direct Parent First*, SDPF. Only the nodes on the paths are in the agent tree; 4) Minimum-energy Multicasting: each source implements minimum-energy multicasting over the agent tree with a more energy-efficient broadcast protocol, IRBOP(B); 5) Dynamic agent tree reconfiguration: When the users move so that the sinks change, the dynamic agent tree can be reconfigured using *Inform Direct Parent Only,* IDPO. Once a stimulus moves, a new dynamic agent tree can be efficiently regenerated over the same RNG with a new source.

Notations:

Energy Path: It is the cumulative energy consumption from the emitter to the receiver (calculation is in Section 2.4.3).

Energy Consumption (*EC*): The energy consumed by a node considers the energy consumption from the root to the node along the energy path. The EC of the receiver is equal to the EC of the emitter plus the energy path from the emitter to the receiver. The EC of the root is zero. Assume a node is u, then its EC is EC(u).

Direct Parent: It is a node's one-hop neighbor in the RNG. When a node looks for its direct parent, it treats its one-hop neighbors as the emitters and recalculates its EC. The one which makes the node's EC be the least will be the node's direct parent.

Two-hop Neighbor: Assume a node A has a neighbor B on the RNG, and B has a neighbor C on the RNG which is not A's neighbor. We define C to be A's two-hop neighbor on the RNG if C is inside A's maximum transmission range, R.

Minimum Transmission Range (R_{tr}): This transmission range is the minimum-energy range large enough to reach a node's one- or two-hop neighbors in the RNG ($R_{tr} \leq R$, calculation is in Section 2.4.3).

2.4.1 Finding RNG Neighbors:

Since sensor nodes are stationary, the RNG neighbor finding can be implemented by off-line. Once the RNG neighbors have been chosen, they will never be changed. A node can find its RNG neighbors as in [2].

In this paper, if a node v is node u's one-hop RNG neighbor, then the edge uv belongs to RNG. We call the induced RNG over the network RNG and the induced MST over the network MST. After a RNG is generated, a node finds its one-hop neighbors in the RNG, exchanges its one-hop neighbor information with its neighbors, and records its two-hop neighbors as in Table 1. After a node records its one and two hop neighbor information, these will never be updated. In the rest of the paper, a node's neighbors means the neighbors in the RNG.

2.4.2 Minimum-energy broadcasting via IRBOP(A)

Table 1: Node and its Neighbors in a RNG

node name	one-hop neighbor	two-hop neighbor
S	A, C, B	A, B, C, D, E, F
A	G, F, S	G, F, E, S, B, C
B	S	S, A, C
C	D, E, S	D, E, F, S, B, A
D	C	C, S, E
E	C, F	C, D, S, F, A
F	A, E	A, G, S, C, E
G	A	A, F

In this section, we explain our *improved RNG broadcast oriented protocol*, IRBOP(A), that is used by the sources to broadcast their position information to all other nodes in the network. In Section 2.4.5, we will discuss IRBOP(B) for sending multicast messages to those nodes (sinks) who are interested in subsequent information that the sources provide.

IRBOP(A) is a minimum-energy broadcasting protocol which is based on the algorithm RBOP proposed in [2]. We summarize RBOP as follows. A source node transmits a broadcast message with the minimum-energy to reach all of its one-hop neighbors in the RNG. Those neighbors apply *neighbor elimination scheme* [13] to decide whether their one-hop neighbors in the RNG received the message or not. Those neighbors respectively forward the message using the minimum-energy to reach all their one-hop neighbors which have not received the message previously. In this way, the message from the source is broadcast to all other nodes. RBOP saves energy using only localized information. However, we have noted that additional savings can be realized with the message sent to two-hop neighbors directly without relaying. In IRBOP(A), a broadcast node needs to calculate its R_{tr} and decide whether to reach its one- or two-hop neighbors. A node which we call the emitter should follow the following rules to determine its *Minimum Transmission Range*, R_{tr}:

A) To guarantee connectivity, the emitter transmits the broadcast message using the minimum-energy to reach all its one-hop neighbors which have not received the broadcast message previously as RBOP does, call this transmission range R_1, and $R_{tr} = R_1$.

B) To save energy, the emitter considers transmission to its two-hop neighbors directly without relaying by its one-hop neighbors. When the emitter finds an one-hop neighbor has not received the message, the emitter sends a HELLO message to the one-hop neighbor, say N, and asks for its R_1, here, we call it R_2. N gives both R_2 and the furthest node name to the emitter where N calculates R_2 based on Rule A. If N's furthest one-hop neighbor, say M, is not the emitter's two-hop neighbor, then the emitter gives up considering two-hop broadcasting in M. If M is the emitter's two-hop neighbor, then the emitter considers transmission to M directly. Let the distance from the emitter to M be R_3. If $R_3^2 < R_1^2 + R_2^2$, then it is more efficient to let R_{tr} be R_3. So, N does not need to forward the message to its one-hop neighbors where the emitter transmits to them directly. The emitter checks each of its one-hop neighbors which have not received the message previously and chooses the maximum R_3 to be its R_{tr} which should be equal or larger than R_1.

Let us consider a graph shown in Figure 2 (To compare with RBOP [2], Figure 2 is from [2]) which consists of RNG edges. Table 1 shows their relations. S is the emitter. S's one-hop neighbors are A, B, and C. S's two-hop neighbors are A, B, C, D, E and F. S calculates its R_1, $\|SA\|$ based on Rule A. Since C is a neighbor and has not received the message (S has not sent it yet), S asks C what is its R_2. C responds with the length of $\|CD\|$ and D. After S finds D is its two-hop neighbor, S calculates $\|SA\|^2 + \|CD\|^2$ and $\|SD\|^2$. Since $\|SA\|^2 + \|CD\|^2 > \|SD\|^2$, D satisfies two-hop broadcasting. Thus, S sets R_{tr} to be $\|SD\|$. Since B does not have any other neighbors, S skips B. S continues to check with node A. Because G is outside transmission range of S, G is not S's two-hop neighbor. S gives up two-hop broadcasting to G. The reason is as follows: We assume C, D, and E have already received the message. We also assume $\|SF\|^2 < \|SA\|^2 + \|AF\|^2$ and $\|AG\| > \|AF\|$. If S applies two-hop broadcasting to F, then S lets R_{tr} be $\|SF\|$. When A receives the message, A still has to forward it to G with $\|AG\|$ and F will receive it also. Thus, the total energy consumption is $\|SF\|^2 + \|AG\|^2$ instead of $\|SA\|^2 + \|AG\|^2$ which is total energy consumption using RBOP (based on the definition of RNG, $\|SF\| > \|SA\|$). Finally, S decides its R_{tr} to be $\|SD\|$.

Let us call the graph induced by R_{tr} on the network to be RNG'. Then, RNG is the subgraph of RNG' where RNG' consists of RNG plus additional edges added by R_{tr}.
Next, we introduce two rules to calculate *Energy Path*:
A) When the broadcast message is from a one-hop neighbor, the energy path is the square of the Euclidean distance between them.
B) When the broadcast message is from a two-hop neighbor, the receiver needs to calculate its energy path from all of the possible energy paths from the emitter to it inside two-hops. The receiver calculates the energy paths from the emitter to the receiver directly which is the square of the Euclidean distance between them, and all of the energy paths from the emitter to its one-hop neighbors plus from the one-hop neighbors to the receiver. The energy path of the receiver is equal to the least one.

For example, in Figure 2, when S is the emitter, the energy path from S to C is $\|SC\|^2$. The EC(C) is equal to EC(S) plus $\|SC\|^2$. When F receives broadcast message from S, F calculates $\|SA\|^2 + \|AF\|^2$ and $\|SF\|^2$. Assume $\|SA\|^2 + \|AF\|^2 > \|SF\|^2$. then the energy path from S to F is $\|SF\|^2$. Thus, EC(F) is equal to EC(S) plus $\|SF\|^2$.

Next, we explain how IRBOP(A) works. The objective of broadcasting from the source is to help the agents find their paths to the source, so, each node can only accept broadcast messages from its one or two hop neighbors. Otherwise, they will not be

Algorithm:

1. The source node emits the message with R_{tr}, which is calculated by R_{tr} calculation Rule A or B. The source node also includes its EC in the message, which is zero.

2. When a node receives a new broadcast message either from its one-hop or from its two-hop neighbors, the node accepts the message:

2.1 The node calculates its energy path and EC.

2.2 The node applies neighbor elimination scheme to eliminate neighbors to whom it does not need to forward the message. The message will not be forwarded if the node's one-hop neighbors have all received it.

2.3 The node checks its one-hop neighbors:

a)If there are some one-hop neighbors have not received the message, then the node calculates its R_{tr} and forwards the message with R_{tr}.

b)If all of its one-hop neighbors have received the message, the node will not forward it.

3. When a node receives a new broadcast message from its non-neighbors, the node ignores the message.

4. When a node receives an already received message:

4.1 The node ignores the message if it has already forwarded it.

4.2 The node applies neighbor elimination scheme to eliminate neighbors to whom it needs to forward the message; the message is ignored if the node's one-hop neighbors have all received it.

4.3 Otherwise, the node calculates R_{tr} and forwards the message with R_{tr}.

Figure 3 IRBOP(A)

able to find their direct parent in the next step using SDPF. As in [2], we apply *neighbor elimination scheme* to one-hop neighbors in the RNG.

For example, in Figure 2, S is the emitter which broadcasts with R_{tr} equal to $\|SD\|$ and includes its EC in the broadcast message (EC(S) is zero). After B receives the message, it checks R_{tr} against the position of its one-hop neighbors. Since S is the only one and is the emitter, B eliminates S, calculates EC(B), and then it stops. After C receives the message, it calculates EC(C) also. Since C deduces that S, E, and D have already received the message, C eliminates them and stops. F accepts the message because S is its two-hop neighbor via A. Then, it calculates EC(F). Since all of its one-hop neighbors have received the message, F will not forward it. D and E do the same as F. A calculates EC(A) and checks its one-hop neighbors. A finds that G is the only one left. A forwards the message to G with R_{tr} equal to $\|AG\|$ and includes EC(A). After G receives the message, it calculates EC(G), checks its one-hop neighbors, eliminates A for transmission, and stops. The broadcast is received by all of the nodes and only two transmissions were needed with a total cost of $\|SD^2\| + \|AG^2\|$.

Figure 3 is IRBOP(A). The cost of the broadcasting is the total energy consumption for sending broadcast messages and the HELLO messages. During IRBOP(A), the complexity of the HELLO message exchange for each node is O(1). Compared with RBOP, IRBOP(A) consumes extra energy in broadcast nodes' one-hop neighbors sending HELLO message to their two-hop neighbors. However, compared to the energy consumed in sending broadcast message, the energy consumed in sending HELLO is little.

2.4.3 Generating the Dynamic Agent Tree

After the source broadcasts its information, all other nodes know where it is. The nodes nearest the users become the sinks (agents). Those agents need to find their paths to the source. We propose a heuristic algorithm, *Shortest Direct Parent First* (*SDPF*), to find the paths from those agents to the source. When each agent finds its path to the source, a dynamic agent tree is generated where the tree is composed of RNG edges.

Shortest Direct Parent First (SDPF): To save energy, each agent should find the minimum-energy path to the source. SDPF is a heuristic algorithm which works as follows:

1) A node looking for its direct parent broadcasts a HELLO message to its one-hop neighbors and asks for their EC. 2) After receiving EC from any of its one-hop neighbors, the node recalculates its EC to the source. The node's updated EC is its neighbor's EC plus the square of Euclidean distance between the one-hop neighbor and the node (energy path between them). 3) The node picks a one-hop neighbor as its direct parent which makes the EC of the node be the least among all its one-hop neighbors. Ties are broken by the node ID. Then, the node sends a HELLO message to inform the neighbor. 4) The direct parent records the node in its children list in the tree. If the direct parent does not have a direct parent, then it goes to step 1. The process continues until the source is found. 5) When all agents find their paths to the source, the dynamic agent tree is completed. The tree consists of all of the agents and their paths to the source, and all edges are RNG edges.

Once a node's direct parent is found, the direct parent will not be changed until the source moves. The cost of generating the dynamic agent tree is the total energy consumption for finding all agents' paths to the source.

2.4.4 Minimum-Energy Multicasting via IRBOP(B)

After the dynamic agent tree has been generated, the source will multicast over the agent tree using the algorithm IRBOP(B). Note that IRBOP(A) is designed to help build the agent tree whereas IRBOP(B) is designed to broadcast efficiently. IRBOP(B) is the same as IRBOP(A) except for the following. In IRBOP(B), a node can accept a message sent from a non-neighbor (i.e., not one- or two-hop neighbor) in the RNG. A node does not need to calculate its EC, and its R_{tr} is calculated as IRBOP(A). When a node receives a broadcast message from a non-neighbor in RNG, as in [2], the node will not forward the message immediately; it will wait until the next time frame (depending on the MAC protocol in use) to decide its R_{tr} and forward the message with R_{tr}. In this way, the source multicasts over the agent tree using IRBOP(B) within the RNG.

2.4.5 Dynamic Agent Tree Reconfiguration

There are two cases in which the agent tree needs to be reconfigured, that is, the users move and the stimulus moves. In the first case, when the users move, some nodes are no longer sinks so they need to leave and be taken off the dynamic agent tree and other nodes become sinks and need to join the tree. We propose an algorithm, *Inform Direct Parent Only* (IDPO), to manage the sinks joining and leaving. In the second case, since the sensor nodes are stationary, the RNG does not need to be regenerated. After a stimulus moves, it broadcasts its position over the same RNG. Thus, a new agent tree can be efficiently regenerated.

Joining (IDPO): When a new user comes, the node nearest to it becomes an agent (sink). The agent does the following to join the existing dynamic agent tree:

1) The sink node looks for its direct parent among its one-hop neighbors in the RNG using SDPF. After finding its direct parent, the node sends a HELLO message to inform it. 2) The direct parent writes down the node in its children list in the tree and looks for its direct parent if it does not have one. 3) The process continues until some direct parent is in the agent tree or the source is reached. Thus, the new path from the new agent to the agent tree is added.

The cost of adding the new sink is the total energy consumption from the time of the new sink sending HELLO messages to find its direct parent to the time of the sink being added to the tree. If a node had been added to the tree, then it can send a HELLO message to its old direct parent directly and inform it of its joining.

Leaving (IDPO): When an user leaves, the sink should be removed from the tree. It does the following:

1) The node sends a HELLO message to its direct parent and informs its leaving;

2) The direct parent (not the root) checks its children list in the tree. If the node is the only one in the children list, the direct parent removes the node from its children list in the agent tree and sends a HELLO message to inform its direct parent. And then, this step is repeated. If the children list is not empty, the direct parent removes the node from its children list but will not inform its direct parent any more.

The cost of deleting an agent (sink) is the energy consumed by nodes sending HELLO messages along the path to be removed from the tree. The complexity of HELLO message exchange for each node is O(1) both in joining and leaving.

3 Performance Evaluation

In this section, we present several simulation results to demonstrate the performance of the minimum-energy tree-based data dissemination scheme, *Dynamic Localized Minimum-Energy Agent Tree-Based Scheme* (DLATS). And, we focus on energy consumption. Our simulations will evaluate three parts: 1) the effectiveness of IRBOP(B); 2) the effectiveness of tree generation with IRBOP(A) and shortest direct parent first (SDPF); 3) the effectiveness of inform direct parent only (IDPO). The simulations are conducted in the following settings: 300 nodes are randomly distributed in a $1000 \times 1000 m^2$. Each value shown in the following figures is averaged over 50 simulations. In our calculations, if the energy consumption includes both broadcast message (data packet) and HELLO message (control packet), we consider both of them.

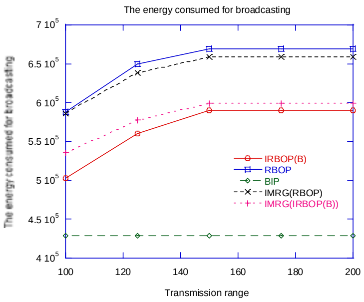

Figure 4 Energy consumption for broadcasting

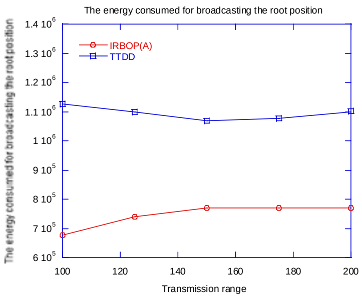

Figure5 Energy for broadcasting root position

3.1 The effectiveness of IRBOP(B)

We randomly pick a node as a source node and let it broadcast to all other nodes once every second for 20 seconds. We vary a node's transmission range from 100 to 200m. Figure 4 shows energy consumed by different broadcast algorithms. To evaluate the effectiveness of IRBOP(B), we compare the energy consumption in IRBOP(B) with RBOP[2], BIP[14], IMRG(RBOP), and IMRG(IRBOP(B)). With IRBOP(B), a node can accept message from non-neighbors in the RNG. RBOP broadcasts to only one-hop neighbors. IMRG(RBOP) means the graph is generated by IMRG [12] and broadcast algorithm is RBOP. IMRG is a topology control algorithm which is used to generate low-weight approximate MST [12]. IMRG(IRBOP(B)) means the graph is generated by IMRG and broadcast algorithm is IRBOP(B). Since BIP is a centralized algorithm, it consumes the least energy and is not affected by the change of node's transmission range. IRBOP(B) consumes less energy than RBOP. This is because IRBOP(B) broadcasts to two-hop neighbors directly when doing so will save more energy than relaying. IMRG(IRBOP(B)) consumes more energy than IRBOP(B). This is because the low-weight approximate MST graph can not guarantee the total energy consumption in the broadcasting is less than RNG. IMRG(IRBOP(B)) consumes less energy than IMRG(RBOP). This is because IRBOP(B) considers its two-hop neighbors in calculation its minimum broadcasting energy.

3.2 The effectiveness of IRBOP(A) and SDPF

To evaluate the effectiveness of IRBOP(A) and shortest direct parent first (SDPF), we compare the energy consumption for: 1) broadcasting a source position using IRBOP(A) with generating grid using TTDD [6]; 2) looking for the paths from the sinks to the root using SDPF with using TTDD; and 3) multicasting over the agent tree using IRBOP(B) where each agent has the optimal path to the root (IRBOP(B, with O)), multicasting over the agent tree using IRBOP(B) where each agent doesnot have the optimal path to the root (IRBOP(B, without O)), with multicasting using TTDD.

Figure 5 compares the energy consumed for broadcasting the source position using IRBOP(A) with building the grid using TTDD. In the figure, TTDD consumes much more energy than IRBOP(A) to inform other nodes the source position. This is because IRBOP(A) broadcasts over a RNG. Figure 6 shows the energy consumed for looking for the paths. We change the number of the sinks (agents) from 1, 5, 10 to 15 in the net-

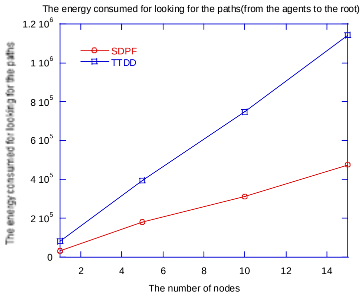

Figure 6 Energy consumed for finding the paths

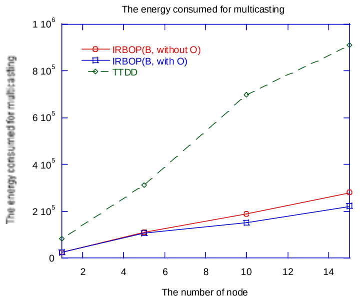

Figure 7 Energy conumed for multicasting

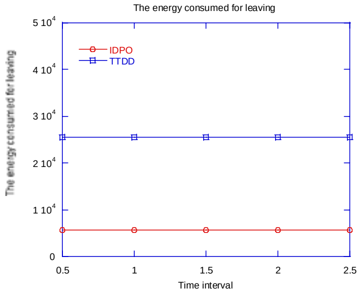

Figure 8 Energy consumed for leaving

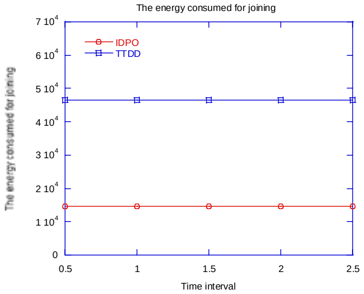

Figure 9 Energy consumed for joining

work. As shown in Figure 6, SDPF saves a lot of energy compared with using TTDD. This is because SDPF is processed in a RNG.

Figure 7 shows the energy consumed for multicasting over the agent tree using IRBOP(B) with optimal path to the root, IRBOP(B) without optimal path to the root, and with TTDD. The number of the sinks is 1, 5, 10, and 15. TTDD consumed much more energy than the two others in multicasting over the sinks. In IRBOP(B, without O), during the building of the agent tree, each agent treats the one-hop neighbor which sends it the broadcast message as its direct parent; if the broadcast message is sent from a two-hop neighbor, then an one-hop neighbor between them will be the node's direct parent. IRBOP(B, with O) consumes less energy than IRBOP(B, without O). When the number of sinks is 1 or 5, the IRBOP(B, with O) consumes around 5% less energy than IRBOP(B, without O).

3.3 The effectiveness of Inform Direct Parent Only

To evaluate the effectiveness of IDPO, we compare the energy consumption of IDPO with TTDD in : 1)leaving; 2) joining.

Figure 8 compares the energy consumed for leaving using IDPO with TTDD. The x-axis is the time interval, which is 0.5, 1, 1.5, 2, and 2.5 m. The number of total sinks in the network is 15 and there is one mobile sink which is moving at speed of 2.5m/s. As shown in the figure, IDPO consumed much less energy than TTDD for leaving. Figure 9 compares the energy consumed for joining using IDPO with TTDD. The x-axis is the time interval which is 0.5, 1, 1.5, 2, and 2.5 m. The total number of sinks in the network is 15 and there is one mobile sink which is moving with 2.5m/s.

4 Conclusion

In this paper, we propose DLATS which can efficiently multicast data from a mobile source to multiple mobile sinks. A dynamic agent tree is generated between each source and multiple sinks. DLATS includes five parts: RNG neighbor finding, source position broadcasting, agent tree generating, source multicasting over the agent tree, agent tree reconfiguration. We propose IRBOP to solve minimum-energy broadcasting over the RNG, which saves energy because of considering two-hop neighbors in broadcasting; we propose SDPF to let a sink find its minimum energy path to the source; we propose IDPO to energy efficiently reconfigurate the agent tree. Simulations demonstrate DLATS outperforms the previous localized solutions for mobile node data dissemination for large-scale sensor networks.

References:

[1] S. M. Das, H. Pucha, and Y. C. Hu, "Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing," IEEE Infocom 2005.

[2] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized Minimum-Energy Broadcasting in Ad-Hoc Networks", IEEE Infocom 2003.

[3] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in Proc. Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc'02), Lausane, Switzerland, 2002.

[4] T. Li, A. Ekpenyong, Yih. F. Huang, "A Location System Using Asynchronous Distributed Sensors", IEEE Infocom, March 2004.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication," Mobicom'00, August 2000.

[6] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," ACM International Conference on Mobile Computing and Networking (MOBICOM'02), p. 148-159, September, 2002.

[7] W. Zhang, G. Cao, and T. La Porta, "Dynamic Proxy Tree-Based Data Dissemination Schemes for Wireless Sensor Networks", IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), 2004.

[8] W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks", IEEE Transactions on Mobile Computing, 2002.

[9] M. Cagalj, J. P. Hubaux, and C. Enz, "Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues", Mobicom'02, Sep, 23-26, Atlanta, Georgia.

[10] J. Wu and F. Dai, "Mobility Management and Its Applications in Efficient Broadcasting in Mobile Ad Hoc Neworks", IEEE Network, July-Aug. 2004.

[11] P. J. Wan, G. Calinescu, and O. Frieder, X. Y. Li, "Minimum-energy broadcast routing in static ad hoc wireless networks", In IEEE Infocom 2001, Anchorage, Alaska, April 2001.

[12] X. Y. Li, Y. Wang, and W. Z. Song, "Applications of k-Local MST for Topology Control and Broadcasting in Wireless Ad Hoc Networks," IEEE Transaction on Parallel and Distributed Systems, Vol. 15, No. 12, pp. 1057-1069, 2004. December 2004.

[13] W. Peng and X. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," in Proc. Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc'2000), Boston, Massachusetts, USA, 2000.

[14] S. Banerjee, A. Misra, J. Yeo, and A. Agrawala, "Energy-Efficient Broadcast and Multicast Trees for Reliable Wireless Communication", Journal of Wireless Networks (WINET) 2002.