Distributed Dynamic Shared Tree for Minimum Energy Data Aggregation of Multiple Mobile Sinks in Wireless Sensor Networks

Kwang-il Hwang, JeongSik In, and Doo-seop Eom

Department of Electronics and Computer Engineering, Korea University,
5-1ga, Anam-dong, Sungbuk-gu, Seoul, Korea
`brightday@final.korea.ac.kr`

Abstract. Sink mobility creates new challenges for several sensor network applications. In mobile sink environments, each sink must propagate its current location continuously, through a sensor field, in order to keep all sensor nodes updated with the direction of data forwarding. This method consumes large amounts of energy. Although several protocols, such as DD, TTDD, and SEAD, have been proposed, in order to solve mobile sink problems, no existing approaches provide both a low delay and energy-efficient solution to this mobile sink problem. In this paper, a distributed dynamic shared tree for minimum energy data aggregation with low delay in highly mobile sink environments, is proposed. In the proposed protocol, the tree is shared with the other slave sinks. Through simulations it is shown that the DST is an extremely energy-efficient, robust protocol with relatively low delay, when compared to DD, TTDD, and SEAD.

1 Introduction

Advances in MEMS, and microprocessor and wireless communication technologies have enabled the development of various applications through the deployment of sensor networks, composed of hundreds or thousands of tiny, low cost nodes.

It is important to note that power is one of the most expensive resources in sensor networks. Due to the difficulty in recharging of thousands of devices in remote or hostile environments, maximizing battery lifetime by conserving power is a matter of paramount importance.

These distributed sensors enable remote monitoring and event detection in a geographically significant region or an inhospitable area. For example, as shown in Fig. 1, explosion area rescuers or robots equipped with handheld devices can obtain dynamic information from sensor nodes thrown over the area. In this paper multiple mobile sinks environments are considered, while sensors are stationary. In the above example, the rescuers or robots may change location, but must be able to aggregate data continuously.

K. Römer, H. Karl, and F. Mattern (Eds.): EWSN 2006, LNCS 3868, pp. 132–147, 2006.

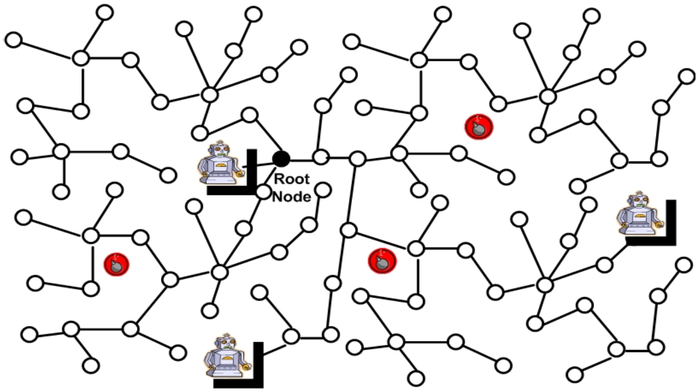


Fig. 1. An example of mobile sink application in sensor networks

Sink mobility formulates attractive challenges for several sensor network applications. Several data aggregation protocols have been developed for sensor networks, such as SPIN [5], Direct Diffusion [2] and GRAB [8]. These protocols can efficiently aggregate data with low delay by constructing one or more aggregation paths on the basis of sink. In-network processing [6] from sources on the paths is also enabled. Nevertheless, these protocols must propagate the sink location continuously through a sensor field in order to keep all sensor nodes updated with the direction of data forwarding.

In order to reduce flooding effect caused by sink mobility, SAFE [3], which uses geographically limited flooding, is proposed. However, in case of highly mobile sinks, local flooding to retrieve the gate connecting itself to the tree increases proportional to the number of sinks in the area.

While the above sink-oriented protocols require continuous reporting of all nodes or paths as sinks move, source-oriented dissemination protocols, such as TTDD [1] and SEAD [4], use mobile sinks' access method to dissemination paths, constructed from each source. Each data source in TTDD proactively builds a grid structure, which enables mobile sinks to continuously receive data regarding the move, by flooding queries within a local cell only. SEAD creates a near-optimal dissemination tree by considering the distance and the packet traffic rate among nodes. Each agent node continuously tracks sink movement. Evidently, these source-oriented protocols perform energy-efficient data dissemination. However, the path per source makes in-network processing impossible. In addition, due to the sinks' access time, more delay is required to aggregate data. No existing approaches provide both a low delay and energy-efficient solution to mobile sink problems.

In this paper, an energy-efficient data aggregation protocol with low delay in highly mobile sink environments is proposed. In order for an aggregation tree to continuously pursue a dynamic sink, *forward sink advertisement* and *distributed fast recovery* is exploited. In the proposed protocol, the shape of the tree is dynamically transformed according to master sink movement, and the tree is shared with the other slave sinks. Therefore, this is called, the Dynamic Shared Tree

(DST) protocol. The DST conserves a considerable amount of energy, despite maintaining robust connection from all sources to sinks, since tree maintenance of the DST is accomplished by distributed local exchanges. In addition, since this represents a kind of sink-oriented tree approach, the DST can aggregate data with low delay along the tree and facilitates in-network processing.

The subsequent sections of this paper are organized as follows. Section 2 introduces the proposed DST protocol. Section 3 describes the maintenance method of the DST to minimize energy consumption. Section 4 illustrates the shared approach of multiple sinks in the DST. In section 5, the energy efficiency of the DST is analyzed in terms of total communication cost compared with DD and TTDD. A comparative performance evaluation through simulation is presented in Section 6. Section 7 concludes this paper.

2 Distributed Dynamic Shared Tree

In this section the basic model of the DST, which is designed to cope well with the highly mobile sink environment, is presented. Then, the DST operation is described in details. The network model for the DST makes the following basic assumptions:

- Homogeneous sensor nodes are densely deployed.
- Sensor nodes communicate with each other through short-range radios. Long distance data delivery is accomplished by forwarding data across multiple hops.
- Each sensor node is aware of its own location (for example through receiving GPS signals or through localization techniques such as [9]).
- Sensor nodes remain stationary at their initial location.
- Sink nodes possess much more energy than that of general sensor nodes, since the battery of a sink node can be recharged and replaced by users.

2.1 Basic Design Concept

The main design goal of the DST protocol is for a tree to continuously pursue a dynamic sink. That means the shape of the tree is dynamically transformed according to the sink's trajectory to maintain a sink-oriented tree as presented in Fig 2.

In the DST protocol, a sink node appoints a Root node. The Root node as an agent of sink becomes an actual root of entire tree, and has an upstream connection to a sink node on behalf of the tree as shown in (a) of Fig. 2. In the distributed DST protocol, the Root node is dynamically changed according to the sink's location, and then nodes in its local area are forced to change their current parent direction to the newly appointed Root node. This DST operation is accomplished by the *forward sink advertisement* and *distributed fast recovery protocol.* For the *forward sink advertisement*, the DST employs a periodic Update Request, which is a periodic local broadcast message. In addition, for *distributed fast recovery*, Root node uses a Sink Lost timer. If a Root node does

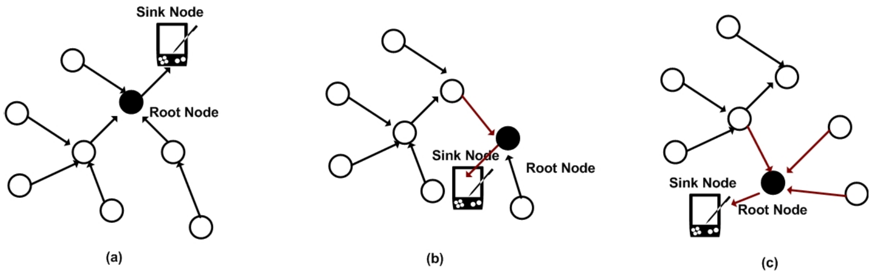


Fig. 2. Description of basic DST operation

not listen to the periodic Update Request message in a given update interval as the sink moves, the Root node's Sink Lost timer expires and the Root node notifies neighbors that the current Root node is disconnected from the sink. If the periodic Update Request interval and time-out value of Sink Lost timer are given by the following Equations (1) and (2), which are related to radio range and maximum sink speed, the current sink's location is discovered by at least one node within two hops of the old Root node, which lost the sink's Update message. Therefore, the nodes, which listen to both the current Update message of the sink and Sink Lost message of the old root, notify its sink that the node itself can become a new root. As soon as the sink receives the notification, it appoints one of them as a new Root (The selection criteria of a new root is arrival time of Root request message). After completing choice of a new root, Root id field in periodic Update message of sink node is changed into new root id. By the changed information, other nodes within sink radio range update parent direction to new root node.

Eventually, according to sink's movement, the original shape (a) of the tree is transformed as shown in (b) and then (c) of Fig. 2. This distributed DST protocol is simple but maintains a robust connectivity to sink. In addition, this distributed approach solves the problems with the excessive energy drain and increased collisions in the traditional sink location update message-based protocols, such as flooding or Directed Diffusion, which require more frequent sink's location update throughout the sensor field. Compared with such sink oriented protocols, the proposed DST save considerable energy more since sink's periodic update message requires the link reversal [7] of only the nodes in the new Root's local area, not entire sensor nodes in the sensor field.

In addition, since the update rates are used with Equation (1), which is proportional to the radio range and inversely proportional to sink speed, the DST can cope well with highly mobile sink environments, and energy waste caused by an excessive update rate can be minimized.

$$Interval = \frac{D}{V_{s-\max}} - (T_p + \alpha) \tag{1}$$

$$Timeout = 1.1 \times Interval \tag{2}$$

$V_{s-\max}$ denotes the maximum sink node speed (m/s), D is a maximum radio range, T_p is a propagation time delay, and α is an additional delay caused by the MAC layer and processing. For realistic interval calculation, (1) is rewritten as follows:

$$Interval = \frac{D}{V_{s-\max}} \times K \tag{3}$$

where K < 1.

Note that the K factor is an important performance parameter for the DST. The optimal K factor is retrieved heuristically by experiment, including the MAC and propagation delay. We choose K = 0.3 as a heuristically optimal value in Section 6.

2.2 Overview of DST Protocol

For distributed operation of the DST in all sensor nodes, each node, except for sink nodes, can have one of four states: Member, Root candidate, Old root, and Root node as shown in Fig. 3.

All the nodes start with a member node state. If a member node listens to the Update Request message with the any specific node's address from a sink, the node's state is changed to the Root candidate state. Each candidate Root maintains a candidate timer during the time receiving the Update request with other node's address. If the sink node moves far from the range of a candidate, the candidate timer expire and the candidate nodes turn to a member state, again. However, if one of them receives the Update Request with its own address, it changes to the Root node state and becomes a new Root node.

A Root node also maintains the Sink Lost timer during receiving the update request with its own address. If the sink moves far from the range of Root node, the Root node cannot hear the next Update request, and therefore the timer expires. As soon as the timer expires, the node changes to the Old root

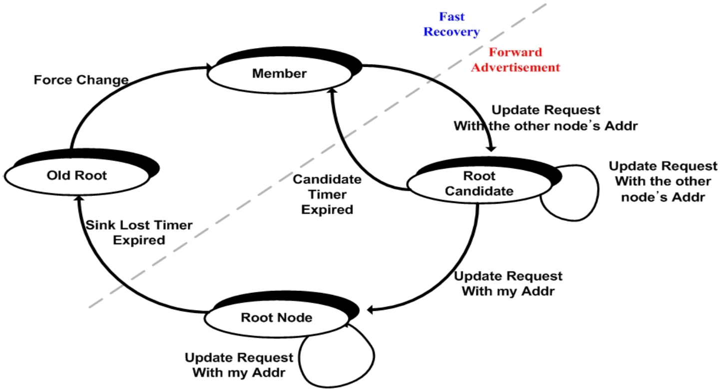


Fig. 3. State transition diagram in each sensor node for distributed DST operation

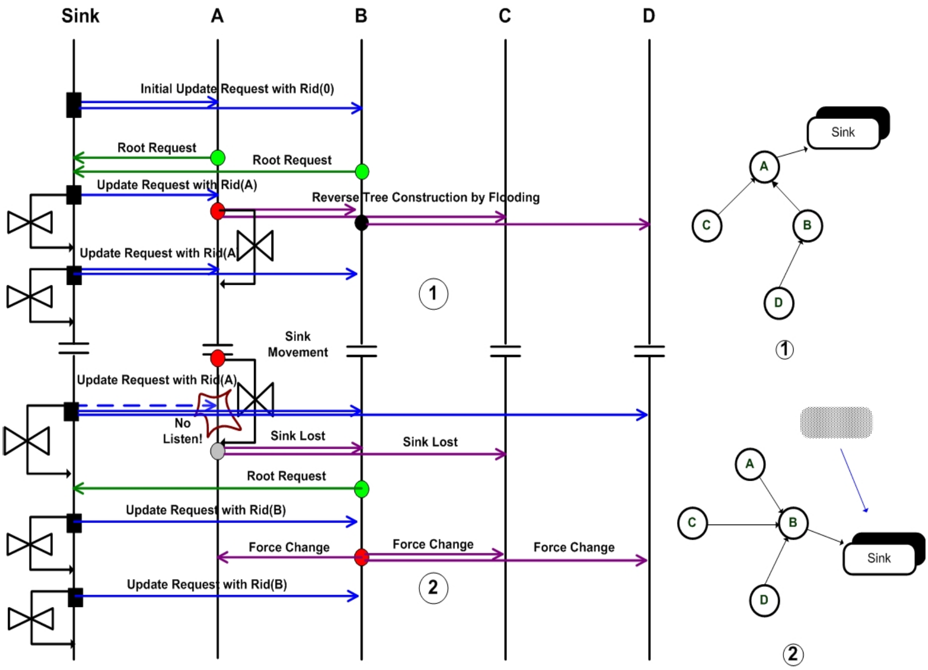


Fig. 4. Message flow in DST operation

state and broadcasts the Sink lost message. The current Root candidate nodes simultaneously transmit the Root Request message to the sink nodes and the only one of candidate nodes is chosen as a new Root node. The newly appointed Root node forces its neighbors to change their parent direction into the new Root node by broadcasting a Force change message. The Old Root node, which hears a Force change message, simultaneously becomes a member node again.

Figure 4 illustrates the message flow for dynamic operation of the distributed DST. Suppose that sensor nodes are initially deployed as in (1) of the Fig. Initially, sink node enters the sensor field, with broadcasting the Update request message periodically. Note that the initial Update request message from the sink is broadcasted with Rood id (0). The nodes, which receive the Update request message from sink, become a Root candidate node and transmit a Root request message to the corresponding sink node as soon as each node receives the Update request with Root id (0). In Fig. 4, the sink node selects node A as a root node and broadcasts the Update request with Root id (A). The node A checks whether the parent already exists and then if it has no parent, which means there is currently no existing tree in the sensor field, it starts flooding. As a result of flooding, a reverse tree as in (1) of the Figure is constructed. While the sink remains within range of the current root, node A, the tree remains stationary without any transforming.

Now, as the sink moves, the DST shows its ability to deal with the mobile sink in earnest. Due to sink's movement, the current Root node, node A, cannot listen to the next Update Request message any more. Instead, node B and D come to listen to the Update Request message from the sink. That means node B and D became Root candidates. Note that since there is no change of Root node by sink, the Root id in an Update request message is still (A). The Sink Lost timer of current root, node A, eventually expires and the node A, which becomes an Old root, broadcasts Sink Lost message. The Sink Lost message of node A can hear only node B so that of the root candidates, node B transmit a Root Request message to the sink.

The sink appoints node B as a new Root node and then restarts broadcasting the Update Request message with Root id (B). As soon as node B receives the new Update message with its own id, it changes the state into Root node and then broadcasts Force Change message. All the nodes, which receives the message, change their parent direction to node B. Eventually, the original tree is transformed, as presented in (2) of Fig. 4. Through this method, the DST can maintain a dynamic aggregation tree only with distributed local exchange of messages.

3 DST Maintenance

3.1 Data Forwarding to Mobile Sink

The DST is intentionally designed to cope well with the mobile sink, so that data forwarding from each source to mobile sink can be easily accomplished along the aggregation tree. The distributed DST protocol can provide seamless data aggregation without concern about the sink's movement, as if the sink had been stationary at the location. The DST outwardly follows a tree-based data aggregation approach so that the data forwarding cost from the source node to sink along the tree is available with $O(\sqrt{N})$. The DST also makes it possible to perform data aggregation combined with In-network processing [6].

3.2 Self-recovery from Partitioned Tree

There are two kinds of situation that cause a partition of a tree.

Failure of fast recovery: This happens when the old Root misses the sink. In this case, the old Root cannot hear the Force Change message of the new root so that the loss of new Root node creates a partitioned tree in which the old Root becomes a root.

Link Failure: All sensor nodes always overhear their own parent's upstream data delivery, except for Root node. When a node, which has upstream data to its parent, cannot overhear its parent node's upstream data forwarding, the node identifies that its sub-tree is partitioned from its parent's tree.

Both of these partitioned trees can be recovered by the *self-recovery process* as follows. Figure 5 demonstrates the self-recovery process of the DST. If the tree is partitioned by some failure factors as presented in (a) of Fig. 5, as soon as a node identifies a failure, the node broadcasts the a Find Root message in the local area as presented in (b) of Fig. 5. Nodes hearing the Find Root message, propagate the message to its upstream along the tree as presented in (c) of Fig. 5. Since the root of a partitioned tree is the sender, itself (problem detection node), the direction of Find Root message propagation is inclined only toward active tree, which is connected to sink as presented in (c) of Fig. 5. In this propagation of Find Root message, each parent avoids transmission of duplicate messages from different children nodes by using a *join request cache*. The reason that the root partitioned tree retrieves only the current Root of active tree, not the other members, is because information from current root is considered as the most reliable.

Eventually, the message reaches the current Root node and the current root node unicasts the PERMIT JOIN message to the root of partitioned tree, through route information obtained from Find Root message as presented in (d) of Fig. 5. Therefore, the Root node of partitioned tree, which received the PERMIT JOIN message, reconfigures its parent's direction to the originator of the message, the current Root node. Finally, as presented in (f) of Fig. 5, the partitioned tree completes the successful joining of the active tree.

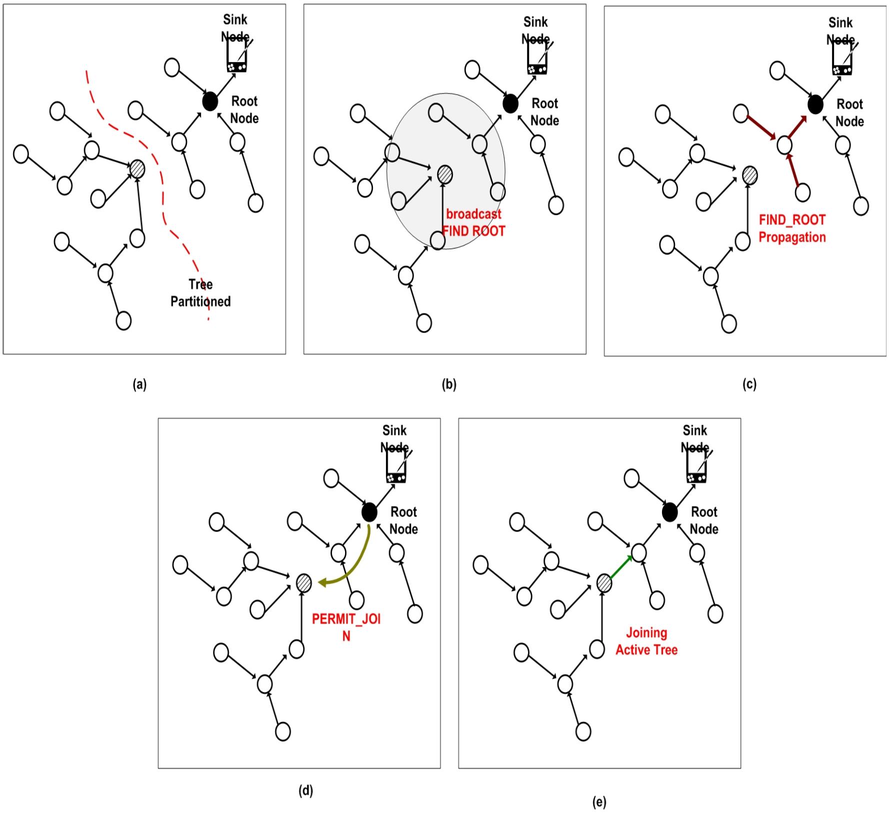


Fig. 5. Self-Recovery process from partitioned tree in DST

4 Multiple Sinks' Share of DST

One of the outstanding features of the DST is the ability to accommodate multiple mobile sinks simultaneously on the same tree. As presented in Fig. 6, multiple mobile sinks share a dynamic tree, which is already constructed by a master sink. Two types of sink are defined in the DST: Master sink and Slave sink. The former is a sink, which is first attended in the sensor filed and directed by root node of the tree. The latter are slave sinks, which join the tree already constructed by a master sink. The master sink must be the only one in a sensor field.

As presented in Fig. 6, slave sinks operate as a leaf node. However, it is different from general member nodes in that the information is transmitted upstream by periodically broadcasting the Periodic Reporting. For the delivery of slave sinks' Periodic Reporting, each node stores multiple sinks' information using a *Sink info cache* table and each parent checks the freshness of Reporting. Each node, which receives the Periodic Reporting message from its child, stores a pair of child id and Slave sink id as well as its sequence number into the Sink info cache. If an identical message is received, the parent ignores the packet. However, if the identical sink id and sequence number received, but from different node, the parent node changes its child id to id of the node of later received message. This is to maintain the freshest routing information in each node. The child ids are used as routing information for downstream data delivery from master to corresponding slave, later. Eventually, the Periodic Reporting message from a slave reaches the master sink node. The master sink transmits the aggregated

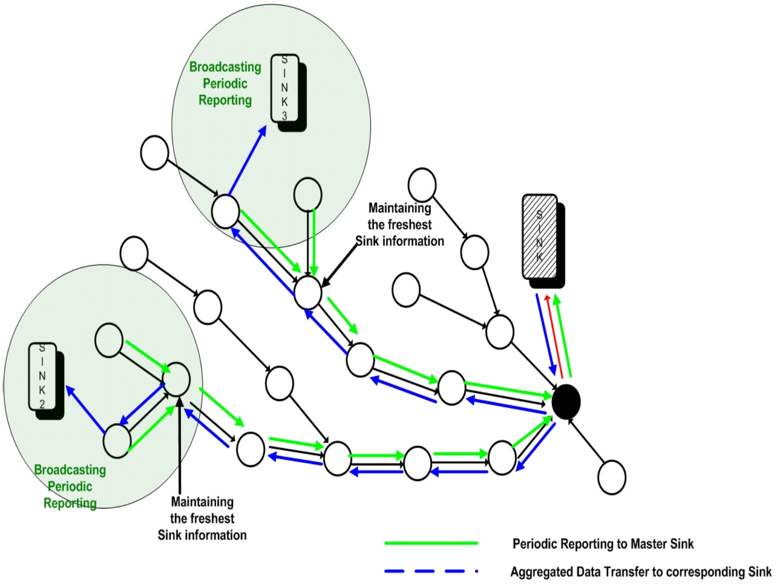


Fig. 6. DST allows data dissemination from each source to multiple mobile sinks on the same tree

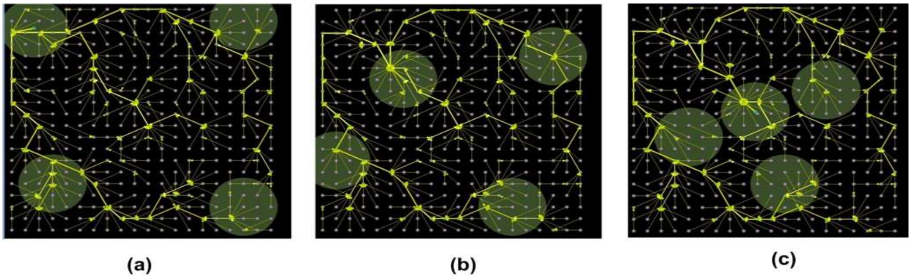


Fig. 7. Dynamic operation of the distributed DST as four sinks move: the operation of the DST conducted on a SHOW_ROUTE program based on the result of NS-simulator

data result to each slave sink through the freshest routing information in the each node's Sink info cache as presented in Fig. 6.

This shared-tree approach of the DST for multiple mobile sinks has advantages described below. The first is to conserve considerable energy. This is because all slave sinks are processed as dynamic leaf nodes on the tree, without requiring additional flooding or each sink's private agents. In addition, by virtue of each parent's route maintenance refresh, each sink's movement is not affected to entire network, but locally processed. The second is that the DST creates a logical star topology between master and slave sinks. Each sink can communicate through a master. In other words, the DST provides a completely two-tiered network structure: the first is sink-oriented tree structure for sensor nodes and the second is a star structure for sink nodes. This structure can facilitate operations of a group of military units on the battlefield.

5 Efficiency Analysis

In this section the energy efficiency of the DST is analyzed. A specific metric is measured: *total communication cost* for data aggregation of the DST, DD and TTDD, respectively.

It is assumed that a square sensor field of area A in which N sensor nodes are uniformly distributed, so that on each side there are approximately $\sqrt{N}$ sensor nodes. There are m event sources and n sink nodes. For grid structure analysis, it is assumed that sensor fields are divided into cells; each cell has αsensor nodes. N_X is defined as the number of cells on the X-axis, and N_Y is the number of cells on the Y-axis.

Directed Diffusion. For event forwarding, Directed diffusion requires four steps: Interest flooding, Exploratory data forwarding, Reinforcement, and Data forwarding. Since Interest forwarding by sink node uses flooding, its cost, in the worst-case, is expressed as mN. The cost for the exploratory data forwarding process to setup multi-paths from source to sink can be approximately given by$\sqrt{N} \times \sqrt{N} = N$. The cost for reinforcement to select single path among the multi-path by exploratory forwarding is also $\sqrt{N}$. Eventually, cost data forwarding along the path is simply given by $n\sqrt{N}$. Accordingly, total communication cost C_{DD} for DD is given by

$$C_{DD} = (m+1)N + (n+1)\sqrt{N} \tag{4}$$

Therefore, in a sink mobile environment, total communication cost is represented as $O(mN)$, including all cost required to construct and maintain a dissemination path and data forwarding with respect to sink movement.

TTDD. TTDD exploits local flooding within the local cell of a grid, which sources build proactively. Each source disseminates data along the nodes on the grid line to the sinks. The TTDD can be divided into three independent steps: Geographical forwarding for grid construction, Query forwarding by sinks, and Data forwarding from sources. Initially, only nodes on the grid line take part in the forwarding process during geographical forwarding. In addition, since the grid is independently constructed by each source, the cost for geographical forwarding is expressed as $n \times N_x \times N_Y \times \sqrt{\alpha}$.

Next, the query is flooded using a sink within a cell and then forwarded along the grid line. Therefore, the cost for query forwarding becomes $\alpha m + m(\sqrt{2N})$. Finally, data forwarding from each source to sink is expressed by $n\left(\sqrt{2N} + \frac{\sqrt{2\alpha}}{2}\right)$. This is because the worst-case sink will be found at the edge cell of diagonal line from source. Eventually, the total communication cost for TTDD is given by

$$C_{TTDD} = n\left(N^{*}\sqrt{\alpha} + \frac{\sqrt{2\alpha}}{2} + \sqrt{2N}\right) + m(\sqrt{2N} + \alpha) \quad , Where N^{*} = N_x \times N_y \tag{5}$$

Therefore, the total cost is $O(m\sqrt{N})$ or $O(n\sqrt{N})$ where $N^{*} << N$, however, TTDD' cost largely depends on the cell size. In addition, in case of many sources, the cost rapidly increases.

DST. Similar to DD, the DST begins with flooding. However, the DST does not require additional flooding with regard to increasing the sinks. Since the tree constructed by flooding, is shared with other sinks, the tree construction cost is only $N + m\sqrt{N}$. In addition, the cost to maintain the tree is negligible, since the DST maintains the tree with some messages exchanged locally when the sink moves. Data is forwarded upstream along the tree and then forward to each slave sink along the tree, so the data forwarding cost is expressed by $(m+n)\sqrt{N}$. Eventually, the total communication cost for the DST is given by

$$C_{DST} = N + (2m+n)\sqrt{N} \tag{6}$$

However, in spite of sink's continuous movement, since flooding is required only in initial tree construction phase, the actual communication cost becomes $O(n\sqrt{N})$, where $m¡n$.

In summary, the DST is similar to the sink oriented data aggregation approaches in shape, however, the DST does not generate any additional flooding or create sizeable communication overhead. Therefore, the energy efficiency for the DST is as good as the TTDD. However, the realistic TTDD' cost largely depends on the cell size as well as the number of sources. More realistic measures are presented with a simulation in Section 6.

6 Performance Evaluation

In this section, the performance of the DST is evaluated through simulations. Simulation metrics and methodology are described in Section 6.1. The main goal in simulating the DST is to evaluate how well it actually conserves energy, maintaining the robust DST connection and low delay in highly mobile environments. The parameters affecting the robustness of DST, are first studied. Then, the performance of the DST is compared to DD, TTDD, and SEAD.

6.1 Metrics and Methodology

The DST is implemented as an independent routing agent module in ns-2.27. In the basic simulation setting, the same energy model is used, which is two-ray ground model and omni-directional antenna, as adopted in Directed Diffusion, and TTDD implementation in ns. A 802.11 DCF is used as the underlying MAC protocol. A sensor node's transmitting, receiving, and idle power consumption rates are set to 0.66W, 0.395W and0.035W, respectively. The network in the simulation consists of 400 sensor nodes randomly or uniformly distributed in a 1000m x 1000m. Each simulation run lasts for 500 seconds. Each query packet is 36 bytes and each data packet is 64 bytes in length, in order to facilitate comparisons with other protocols.

Three metrics are used to evaluate the performance of the DST.

Success rate is the ratio of the number of successfully received data at a sink from source. This metric demonstrates the robustness of the aggregation path.

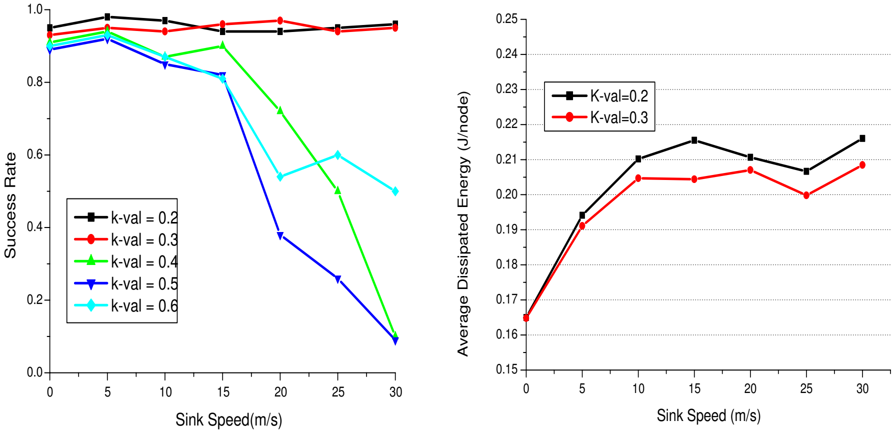
Average end-to-end delay is measured by averaging delay from source to sinks. This metric indicates the freshness of data packets.

Energy consumption per node is defined as the total communication energy the node consumes. The communication energy includes tree construction, data dissemination, and sink mobility management.

6.2 Robustness of DST

We first study on a parameter affecting the performance of DST. In Section 2, we already emphasized the impact of k-value which is a parameter determinig period of Update Request of sink for continuous connection of the dynamic tree according to sink's mobility.

Figure 8 (a) demonstrates the success rate at different sink speeds. Note that the success rate with k-val more than 0.4 is drastically deteriorates around a sink speed of 15 m/s. This is because the breaking probability of the tree is increased



(a) Success rate of DST v.s. Sink Speed

(b) Average dissipated energy v.s. Sink Speed

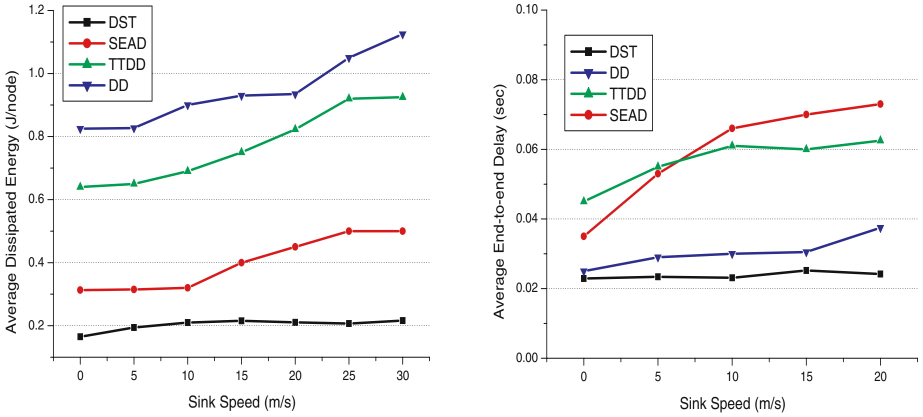
Fig. 8. Robustness of the DST

with a large k value. However, it can be shown that the DST maintains a success rate more than 0.9 with a k value of 0.2 and 0.3 in spite of the high mobility of sink. Prior to choosing the best k value by the heuristic result of this experiment in (a), another observation related to the k value is made presented in (b) of Fig. 8. This represents the energy consumption of each k value, 0.2 and 0.3. Intuitively, we know that the larger k value, the more energy increases. The result in Fig. 8 (b) proves this fact. Consequently the two observations provide the most efficient value for k, 0.3, for both robustness and energy conservation.

6.3 Impact of Sink's Mobility

In this subsection the impact of sink mobility on the performance of the DST, is evaluated. In the simulation setting, 8 mobile sinks and 30 randomly chosen sources, in the sensor field, were used. Energy consumption and average end-to-end delay according to varying the maximum speed of a sink, are measured from 0 to 30 m/s.

Figure 9 (a) presents the average dissipated energy as the sinks' speed is varied. In this Fig, the DST presents superior energy consumption over the other protocols. This is because the DST can maintain the aggregation tree dynamically as the sinks migrate. In addition, the DST does not require additional flooding or location notification to access nodes or agents. In DD, the entire topology is changed so the new location of the mobile sink is propagated throughout the sensor filed in order for all nodes to obtain the sink's location. The TTDD is designed for mobile sinks, but cannot avoid rebuilding a new multi-hop path between the sink and the grid to track the sink's location. Although SEAD based on the source-oriented steiner tree shows smooth energy increase, SEAD has the overhead that each sink must recognize their specific location to continuously access nodes or to change access nodes.



(a) Energy consumption v.s. Sink Speed

(b) Average delay v.s. Sink Speed

Fig. 9. Impact of sink's mobility

Since the DST only uses local interactions on the basis of the sink to maintain dynamic tree, the increase of energy consumption in terms of the entire sensor network is considerably moderate, as presented in (a) of Fig. 9.

Figure 9 (b) presents the average end-to-end delay as the sinks' speed is varied. This Figure demonstrates that DD and DST maintain relatively lower delay than TTDD and SEAD. TTDD and SEAD are source-oriented data dissemination protocols so that they require finding a valid path from source to each sink, whenever a source generates an event. Finding the valid path adds extra delay to the protocols. Conversely, the DST and DD, which are sink-oriented approaches, do not require such additional delay, since all sensor nodes already know the path to each sink. Nevertheless, as a sink's speed increases, DD demonstrates gradual increase in delay. This is because of the flooding effect according to sink mobility. However, the DST which maintains a dynamically dissemination tree oriented to sink, shows an almost constant delay variation as presented in (b) of Fig. 9.

6.4 Impact of the Number of Sinks

In this subsection, the impact of the number of sinks on the performance of the DST, is evaluated. In this simulation, the sinks' speed is set at 10 m/s and energy consumption is measured as the number of sinks increases to 8.

Figure 10 presents the energy consumption as the number of sinks is varied. This Fig demonstrates that in case of a single sink TTDD and SEAD outperform the sink-oriented protocols, the DST and DD. This is because the DST requires basic energy consumption to maintain the tree using the periodic Update message. However, as the number of sinks increases, energy consumption in the DST only slightly increases, in contrast to the other protocols. This is because the dynamic tree is shared with the other multiple sinks. As a result, there is little additional energy per sink, in contrast to the other protocols, such as DD, TTDD, and SEAD.

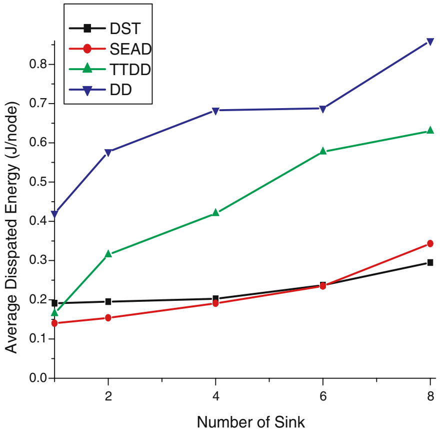


Fig. 10. Energy consumption v.s. number of sinks

7 Conclusions and Future Work

In this paper, an energy-efficient data aggregation protocol with low delay in highly mobile sink environments in sensor networks, is proposed. In order to continuously maintain aggregation tree, a *forward sink advertisement* and *distributed fast recovery,* was utilized. In the proposed protocol, the tree is shared with the other slave nodes so that it is called the Dynamic Shared Tree (DST) protocol. Through simulations, we showed that the DST is a considerably energy-efficient, robust protocol with low delay, compared to Directed Diffusion, TTDD, and SEAD, in highly mobile sink environments.

The DST is currently being investigated on a large-scale sensor network test-bed.

References

1. Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, Lixia Zhang, "TTDD: Two-tier Data Dissemination in Large-scale Wireless Sensor Networks," *ACM/Kluwer Mobile Networks and Applications, Special Issue on ACM MOBICOM*,2002.
2. C. Intanagonwiwat, R. Govindan, and D. Estrin "Directed diffusion for Wireless Sensor Networking," *IEEE/ACM Transaction on Networking*, Vol. 11, 2003.
3. Sooyeon Kim, Sang H. Son, John A. Stankovic, Shuoqi Li, Yanghee Choi, "SAFE: A Data Dissemination Protocol for Periodic Updates in Sensor Networks," In *Proceedings of the 23 rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)*,2003.
4. Hyung Seok Kim, Tarek F. Abdelzaher, Wook Hyun Kwon "Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," In *Proceeding of Embedded Networked Sensor Systems (SenSys03)*, Los Angeles, California, USA, 2003.

5. W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks,” *ACM International Conference on Mobile Computing and Networking (MOBICOM’99)*, 1999.
6. B. Krishnamachari, D. Estrin, and S. Wicker, “The Impact of Data Dissemination in Wireless Sensor Networks,” In *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, 2002.
7. C. Busch, S. Surapaneni, and S. Tirthapura. “Analysis of Link Reversal Routing Protocols for Mobile Ad Hoc Networks,” *SPAA 2003*, pp. 210-219, San Diego, California, June 2003.
8. F. Ye, S. Lu, and L. Zhang. GRAdient Broadcast: A Robust, Long-lived Large Sensor Network.
9. K. Langendoen and N. Reijers, “Distributed Localization in Wireless Sensor Networks,” Computer Networks (Elsevier), special issue on Wireless Sensor Networks, August, 2003.