Adaptive Reversal Tree Protocol with Optimal Path for Dynamic Sensor Networks

Kwang-il Hwang* ***Lifelong Member***

ABSTRACT

In sensor networks, it is crucial to reliably and energy-efficiently deliver sensed information from each source to a sink node. Specifically, in mobile sink (user) applications, due to the sink mobility, a stationary dissemination path may no longer be effective. The path will have to be continuously reconfigured according to the current location of the sink. Moreover, the dynamic optimal path from each source to the sink is required in order to reduce end-to-end delay and additional energy wastage. In this paper, an Adaptive Reversal Optimal path Tree (AROT) protocol is proposed. Information delivery from each source to a mobile sink can be easily achieved along the AROT without additional control overhead, because the AROT proactively performs adaptive sink mobility management. In addition, the dynamic path is optimal in terms of hop counts and the AROT can maintain a robust tree structure by quickly recovering the partitioned tree with minimum packet transmission. Finally, the simulation results demonstrate that the AROT is a considerably energy-efficient and robust protocol.

Key Words : Link Reversal, Mobility Management, Routing Protocol, Optimal Path, Tree Management, Wireless sensor networks

Ⅰ. Introduction

Recently sensor networks have led to tremendously incremental utilization in many military and civil applications such as battlefield surveillance, environmental control, and security management. In such applications, it is crucial to reliably and energy-efficiently deliver sensed information from each source to a sink node. However, if a sink moves to the other place, the dissemination path may no longer be effective. The path will have to be continuously reconfigured according to sink movement. The continuous re-configurations can bring about tremendous traffic and energy wastage inthe network. In particular, energy is considered as one of the most expensive resources in sensor networks. In addition, path optimality from each source to a dynamic sink is regarded as another important design factor in sensor network protocol design space, since the optimal path reduces the end-to-end delay and additional energy wastage.

In order to efficiently manage the dissemination path in sensor networks with a mobile sink, several schemes are proposed [2 -5 and 6]. Directed Diffusion [3], SAFE [4], and DST [6] concentrate on managing the dissemination path initiated from a sink, whereas source initiated dissemination protocols, such as TTDD [2] and SEAD [5], use a method that mobile sinks access on the dissemination path constructed on the basis ofeach source.

Although these methods aim to solve the problem using different techniques, high

---

※ 본 연구는 2007년도 시립인천전문대학 교내연구비 지원에 의하여 수행되었음

* 시립인천전문대학 컴퓨터제어과 (brightday@icc.ac.kr)

논문번호 : KICS2007-03-124, 접수일자 : 2007년 3월 19일, 최종논문접수일자 : 2007년 10월 2일

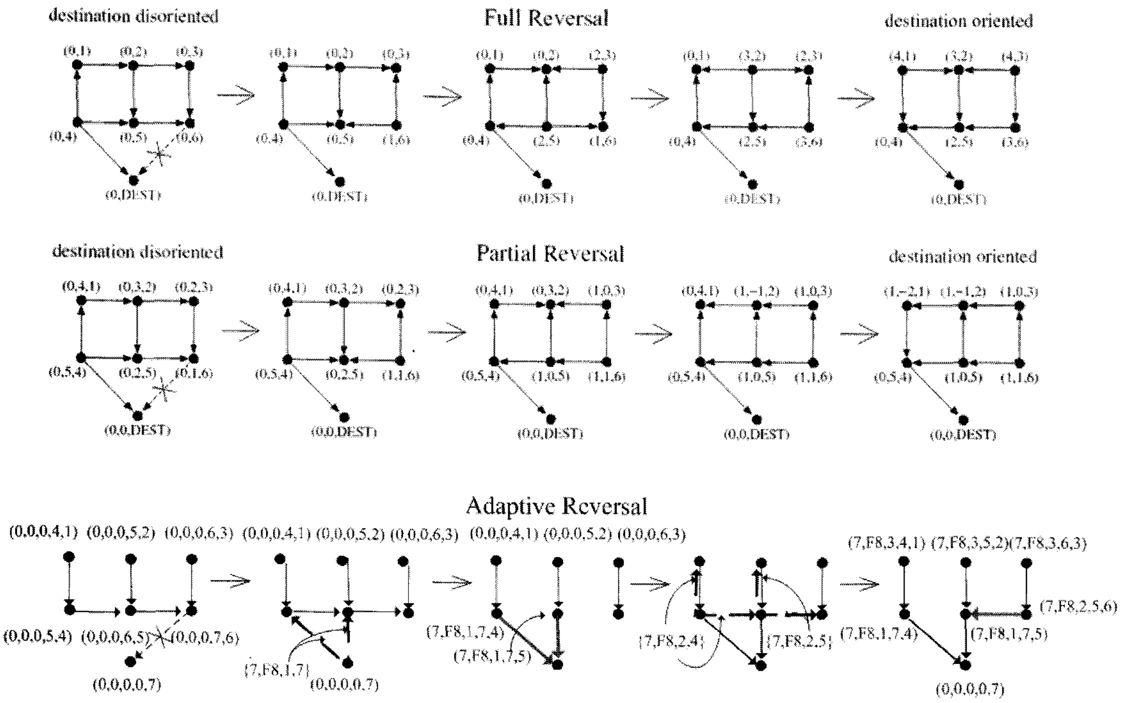


Fig. 1 GB full and partial reversal and adaptive reversal with optimal-path algorithms

maintenance overhead is still required to continuously update the dissemination path. Furthermore, dissemination paths that they manage do not guarantee the optimal path.

In this paper, to address these several challenges on the mobile sink problems, an *Adaptive Reversal Optimal path Tree* (AROT) protocol is proposed. The AROT is able to maintain the dynamic tree with optimal path by adaptively reversing links with minimum communication overhead in the network.

The remainder of this paper is organized as follows. Section 2 overviews the basic algorithms for the AROT. Section 3 describes detailed operation of the AROT including sink mobility management, data dissemination, and error recovery. Section 4 evaluates the optimality of AROT. In section 5, performance evaluation through simulation is presented. Section 6 concludes this paper.

Ⅱ. Overview of the Proposed Algorithm

Link reversal algorithms provide a simple mechanism for routing in mobile ad hoc networks with dynamic topology changes. Gafni and Bertsekas [1] proposed full and partial reversal algorithms. In the link reversal algorithms, a directed acyclic graph (DAG) directed at the destination is continuously maintained according to topology changes. However, such classical reversal algorithms are hardly considered a routing protocol for sensor networks, due to various peculiar characteristics of sensor networks. First, while all nodes in a MANET can become a source or sink, all flows of data in sensor networks are eventually concentrated on the sink node. In addition, the majority of sensor nodes are stationary and topology changes result from the movement of the sink node, rather than sensor nodes.

In particular, in traditional link reversal routings

(LRR), the invocation of reversal is initiated from nodes detecting an absence of destination. However, in sensor networks where innumerable sensor nodes are densely deployed and communicate with each other in an ad hoc manner, such an update reversal triggered from the node detecting absence of the sink due to sink movement can be extended to the entire network, because each node in the network can hardly identify the current location of the sink. Eventually, due to continuous sink movement, link reversal routing can result in continuous flooding in the network.

Therefore, in order to address such problems, an *Adaptive reversal with Optimal-path* (ARO) algorithm is proposed for sensor networks with a mobile sink. The ARO algorithm has several distinguished characteristics. First, while the traditional LRR maintains a DAG for each destination at each node, the ARO is based on a spanning tree directed at a temporary root in which multi paths are not permitted. Secondly, update reversal invocation is triggered from the node appointed by the sink node, not the node detecting absence of sink due to sink movement.

The ARO is implemented in a straightforward manner by a cache maintained by each node, and an update reversal message triggered from the node that wants to be a new root in the network, which is appointed by the sink node. The cache is called the Uplink_info cache. The Uplink_info cache of each node i is the set of quintuple (a_i, b_i, c_i, d_i, i), ordered lexicographically. Essentially, the field a_i is a temporary root node id, b_i is a sequence number received from latest update message, c_i represents the distance from the temporary root (i.e. hop distance), d_i designates the current parent node, and i is its unique id in the network. Let N denote the set of nodes in the network. The initial set of Uplink_info cache $\{(a_i^0, b_i^0, c_i^0, d_i^0, i) | i \in N\}$ satisfies $a_i^0 = b_i^0 = c_i^0 = d_i^0 = 0$. A temporary root node constructs a spanning tree by flooding. It is assumed that the tree is directed to the temporary root and each node considers only one parent node to reach the root. Accordingly, after constructing the spanning tree starting from temporary root node k, where $a_k = k \; and \; c_k = d_k = 0$, the set of Uplink_info cache for each node i is changed as follows: $\{(a_i^k, b_i^k, c_i^k, d_i^j, i) | i, j, k \in N, c_j^k \langle c_i^k, and \; d_i^j = j\}$. If a node l wants to become a new root, it propagates the update reversal message with the set of quadruple $\{(a_l, b_l, c_l, l) | l \in N\}$, where a_l, b_l, c_l and l have the same meaning as $a_i, b_i, c_i \; and \; i$ in the Uplink_info cache, respectively. The link reversal process of the AR algorithm is implemented as follows:

Suppose node i received the update packet from node j. Each node i compares each field in its Uplink_info cache with the update packet from j, including the set of (a_j, b_j, c_j, j). Let the set $(\bar{a}_i, \bar{b}_i, \bar{c}_i, \bar{d}_i, i)$ denote the updated cache after the reversal process.

First, the field d_i is compared. If d_i is equal to j, (i.e., j is already the parent), then the cache in node i is updated with $\bar{a}_i = a_j, \bar{b}_i = b_j, and \; \bar{c}_i = c_j + 1$. It is important to note that, in this case, the node does not propagate update messages to other nodes any more.

On the other hand, if j is different from d_i, according to other conditions, the reversal process of each node goes into one of the following:

Case 1: $a_i \neq a_j \; and \; b_i \neq b_j$ (the node should forward the update reversal). The cache is updated with $\bar{a}_i = a_j, \bar{b}_i = b_j, \bar{c}_i = c_j + 1, and \; \bar{d}_i = j$ and the node propagates its update message $(\bar{a}_i, \bar{b}_i, \bar{c}_i, i)$.

Case 2: $a_i == a_j \; and \; b_i == b_j$

If $c_i \leq c_j$, the node does nothing, otherwise, $\bar{c}_i = c_j, and \; \bar{d}_i = j$. The latter case means that the link of the node $i$for this temporary root has already been reversed, but a more efficient route to the root is discovered so the new path is selected. As a result, after each reversal process, all of the nodes maintain optimal path to the sink in terms of hop counts.

Figure 1 describes the reversal process of the ARO algorithm. It is remarkable that the update reversal packet in the ARO is only forwarded to the partial area around the new root node, not to the entire network. As shown in fig. 1, if the other nodes are more connected to the back of nodes 1, 2, and 3, the update reversal packet will be no more be propagated. This reduced amount of update packets can lead to considerable energy conservation in the networks in spite of guaranteeing the fresh optimal path. In section 5, it is shown that maintaining the dynamic spanning tree is available with the number of nodes less than approximately 14% of all nodes, in spite of a highly mobile sink environment.

Ⅲ. Adaptive Reversal Optimal-path Tree (AROT)

In this section, the detailed operation of *Adoptive Reversal Optimal-path Tree (AROT)* is presented. Since the AROT proactively performs adaptive sink mobility management, data dissemination from each source to a mobile sink can be easily achieved along the AROT, without additional control overhead. In addition, the AROT can maintain a robust tree structure by quickly recovering the partitioned tree with minimum packet transmission.

3.1 Basic model

The application model presented is somewhat different from general sensor network applications which take a stationary sink node into account as a whole. The application model, in particular, focuses on mobile sink applications where a sink node enters the sensor field directly and performs roles based on source data dynamically refreshed from a sensor field. The application model is very useful for battle field or rescuer activity.

The application model assumed also 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.
- Sensor nodes remain stationary at their initial location.

3.2 Sink mobility management

At the initial stage, a sink node enters the sensor field, with broadcasting the Sink_Update. The update message is used to notice the current sink's location to the network and thereby a temporary root node is changed to a new node that is currently near the sink. In general, the Sink_Update message is periodically broadcasted by the update rates, which represents the time interval for a user to obtain information sensed from the sensor field.

However, it is also allowed that the update request of the sink is broadcasted asynchronously (Non-periodically) only when the sink wants.

All nodes that received the Sink_Update message transmit the Root_Request message to the sink node. The sink appoints a new sink as a new root node and replies with the Root_Reply to the node. However, if one of these nodes is the previous root, the sink does not change the root, because this means the sink has not moved far away from the root since the last update. If the *tree construction field* in the Root_Reply message is set, (i.e. the sink requests a new dissemination tree), the root node initiates construction of an initial spanning tree by flooding the entire network. The tree is directed to the current root node.

Fig. 2 illustrates an example when managing the sink mobility of the AROT. As the sink moves to a new location, a new root node is appointed and the adaptive reversal process is triggered from the node. In the AROT, the *stable zone* and the *reversal zone* are defined as the area where update activity does not occur, and as

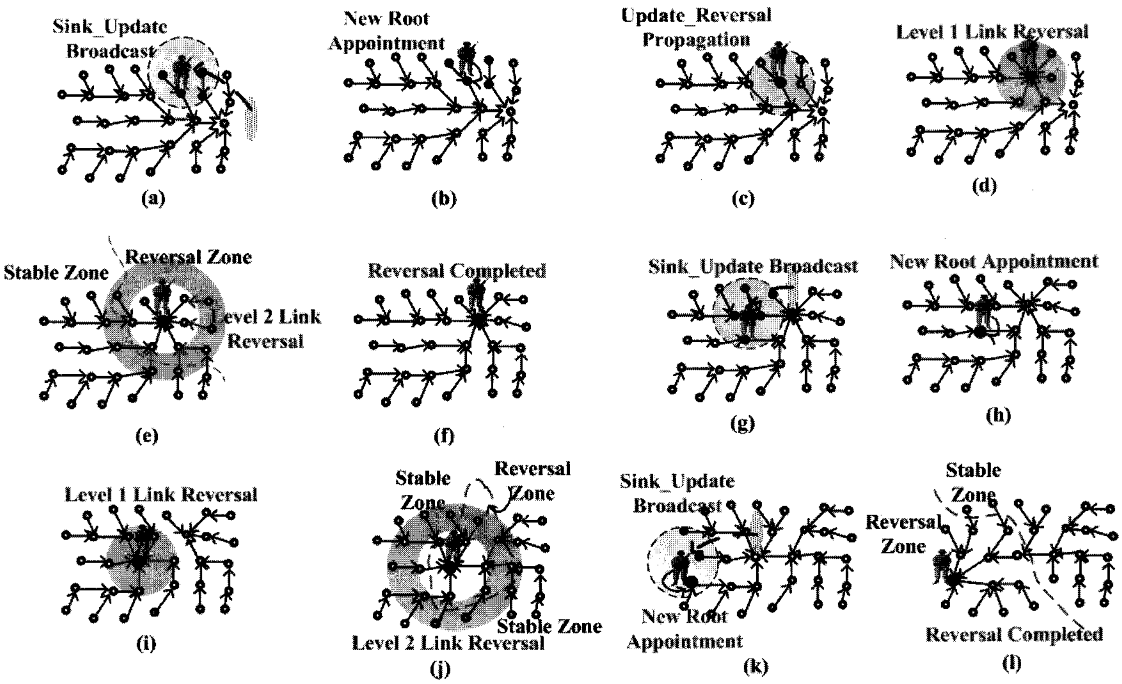


Fig. 2 Example of adaptive sink mobility management via the AROT

the area where reversal of each node is updated, respectively. The update reversal process for each update request proceeds until update propagation meets the stable zone. It is important to note that in spite of sink movement, update reversal is only required in a partial area (reversal zone), not the entire network. The size of stable zone and update rate interval represents a trade off. The stable zone presents the set of nodes which are already correctly directed to the new root node. On the other hand, the reversal zone stretches out from an area, where the last update was invoked, to that adjacent to the current root node. Accordingly, in the case of a long update interval, a sink may move further away from the old root, so that it leads the extension of the reversal zone.

In section 5, the working node rates are investigated, revealing the ratio of reversal zone over the entire field, with respect to the variation of sink speed.

3.3 Proactive data dissemination via AROT

Since the AROT manages sink mobility proactively, data of each source is simply disseminated upwardly along the tree as other tree-based dissemination protocols are. The two types of data model can be applied in AROT: *one-time event* and *periodic reporting* of detection node. In particular, the latter model is useful for applications such as mobile target tracking. In this model, the node that detects target should disseminate event to the sink with specific update rate during the time when a target residues there. By collecting the data from each event source, the sink can identify current location and trajectory history of the target. Event update rate is given by the sink when an interest (query) is flooded. Remember that the AROT utilizes *reversal update* by sink periodically for the path freshness of the tree. In the periodic reporting data model, if the reversal update rate is longer than update rate of sources, freshness of the data will be deteriorated as well as delay will be increased . Thus, the reversal update rate should be taken by smaller value than event update rate. This can enables the data dissemination with low delay along the tree.

Figure 3 illustrates data dissemination process from a source to the sink. Every node on the path from the source to the sink has the responsibility to reliably deliver the data to the sink. Therefore, Data and ACK mechanism between each link is used as shown in fig. 3(a). If a node meets reversal process on the way to deliver, the node in the reversal zone waits until the reversal process is completed (new path is reconstructed). Then, the data dissemination process is restarted from that node to sink as shown in fig. 3 (c).

As a result, the data dissemination cost from each source to mobile sink can be accomplished with the $O(n\sqrt{N})$ along the tree, where n is the number of source nodes and N is the number of deployed sensor nodes. In the case of many sources, the cost will be reduced greater by use of in-network processing.

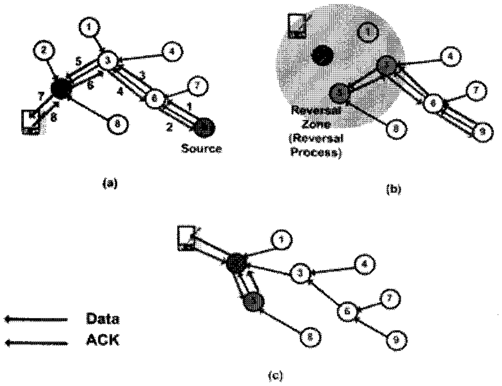
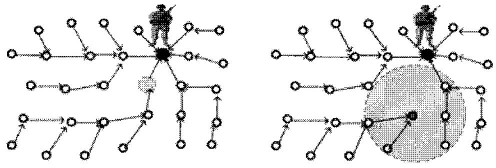


Fig. 3 Data dissemination along the AROT: each node on the path utilizes ACK

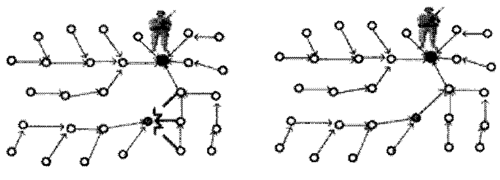
3.4 Error recovery to failures

A link on the tree can be broken by various reasons, such as node failures, obstacles, and so on. Such link breakages result in partitioned trees. However, it is important that the tree should always maintain robust connectivity from all sensor nodes to sinks. Therefore, in order to keep the robust tree, the AROT exploits simple but fast, robust error-recovery.

As mentioned in the pervious subsection, each data sent by a child is acknowledged by each parent. By this acknowledge, we can identify the status of the link whether the tree is partitioned or not. As soon as the root of a partitioned tree comes to realize the breakage of its uplink, it broadcasts a Join_Request message to its neighbors. Only nodes, where their parent is not the sender, reply with the Join_Reply to the root of the partitioned tree. The node completes joining the active tree by selecting one of them as its parent. The selection condition is the node having the smallest height to current root node of active tree as shown in Fig. 4.



(a) Node Failure (b) Join Request

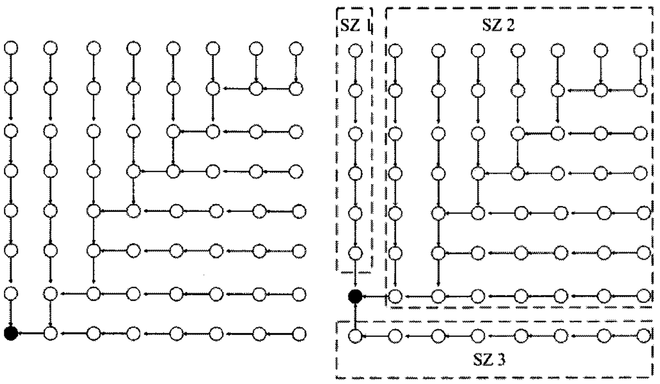


(c)Join Response (d) Join Complete

Fig. 4 Error recovery process of the AROT for a partitioned tree due to failures

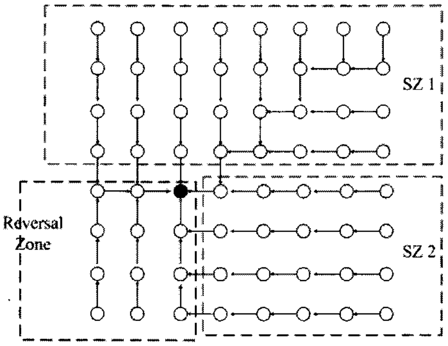
Ⅳ. Optimal Path Analysis

Due to the sink mobility, the shape of the dissemination path is transformed from original one. Some protocols such as DST only focus on the energy efficiency in energy efficiently maintaining the dissemination path in dynamic mobile sink. Therefore, the optimal path is not guaranteed at any instance. In addition, the path tends to get worse as far as the sink moves continuously.



(a) Initial ART

(b) Optimality of Stable Zone



(c) Optimality of Reversal Zone

Fig. 5 Path optimality of AROT

In particular, we mainly focus on the hop count cost for the optimal path since the hop distance can reflect each physical distance between sources and sink under the assumption that sensor nodes are densely deployed. This means the shorter path, the lower is delay. In addition to this, this shortest path can provide the physical shortest path to the target. Therefore, the optimality of dissemination path is one of the most important performance factors. In this Section, we analyze a path optimality of AROT.

It is assumed that a square sensor field of area A in which N sensor nodes are uniformly distributed as shown in fig. 4.6, so that on each side there are approximately $\sqrt{N}$ sensor nodes. We denotes each source and sink i and s, where $\forall i \in N$. It is also assumed that all nodes have the same radio range and the range is limited in the distance including only each node's neighbors (left, right, up, and down). This means a neighbor located at diagonal direction is not included.

Theorem 1 All of the sources, $i \in N$, in the AROT constructed by initial flooding have optimal path to the sink.

Proof By assumptions mentioned above, the network plane can be coordinated to (x, y) and we also assume that the sink is located at the left corner as shown in fig. 5(a). As validated in that figure, route from an arbitrary node with (x, y) on the tree to the sink along the tree is the shortest path. This means that all the routes on the tree present the minimum hop count value in all the routes that are reachable from a arbitrary point (x, y) to the sink (0, 0). In addition, the minimum hop count for (x, y) is x + y.

For example, there is no route from node (3, 3) to the sink (0, 0) with less than 6 and the minimum hop count for node (7, 7) is 14.

Lemma 1 Let N be a full stable AROT. If F is a sub-tree of N rooted at a node f, i.e., $F \subset N$, all the routes along the tree from each source, $i \in F$ to the root f have optimal path.

Proof Suppose a tree in which each node has only one child, respectively. The tree L can be represented by the following set, $L = \{l_1, l_2, l_3, l_4, .. l_k\}$. By assumption, each pair consisting of a parent and a child is one hop. So that all sources in F, a sub-tree of L, rooted at l_i, for example $F_3 = \{l_3, l_4, .. l_k\}$, have optimal path to the l_i.

Theorem 2 All of the routes along the AROT from each source to sink are optimal after reversal process is completed.

Proof If a sink node moves to (0, 1) and requests adaptive reversal, all neighbors around the sink changes their parent direction to the sink node. Then, they propagate the reversal request. However, as shown in fig. 5 (b), the reversal is not propagated any more from their current neighbors. This is because the rest nodes are already directed to its parent. As mentioned in the previous section, the area where reversal process regarding current reversal request does not occur is called stable zone. As shown in figure 5 (b), the stable zone with respect to current reversal request can be divided into the three sub zone, SZ1, SZ2, and SZ3. Let be t current request time. If the tree was optimal regarding all the sources at the last request time t-1 as in fig. (a), each stable zone at current request time $t$is a sub-tree of (a). Accordingly, by lemma 1, all the stable zones, SZ1, SZ2, and SZ3, are optimal tree. In addition, since all the sub-trees are connected to the sink with one-hop distance, all of the routes along the AROT from each source to sink are optimal even after reversal process is completed as shown in fig 5 (b).

Now suppose that the sink moves to (2, 4). By algorithm, the reversal process at request time $t + 1$ creates two partial stable zone, SZ1 and SZ2, and one reversal zone as shown in fig. 5 (c). Reversal process finds shorter path from its own to current sink by comparing distance in the reversal request message received from its neighbors. In the end, the after completing the current reversal process, all the sources in the reversal zone have optimal path to current sink. Consequently, the whole tree consisting of stable zone and reversal zone is optimal as shown in fig. 5 (c).

Ⅴ. Performance evaluations

In this section, the performance of the presented AROT is evaluated in terms of various metrics, including success ratio, average end-to-end delay, communication overhead and dissipated energy.

5.1 Methodology

The presented AROT 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 a two-ray ground model and omni-directional antenna, as adopted in Directed Diffusion, and TTDD implementation in ns. 802.11 DCF is used as the underlying MAC protocol. A sensor node's transmitting, receiving, and idling power consumption rate is set to 0.66W, 0.395W and0.035W, respectively. The network in the simulation consists of 400 sensor nodes randomly distributed in 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.

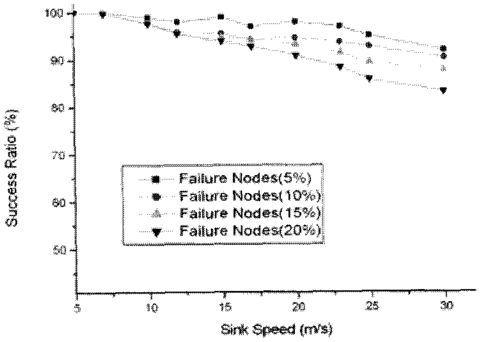


Fig. 6 Success ratio vs. Sink speed

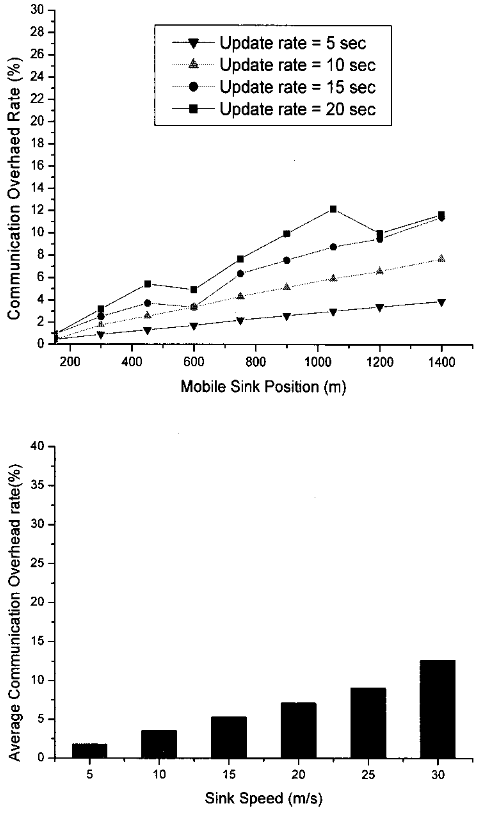


Fig. 7 Evaluations of communication overhead

5.2 Robustness of AROT

In this subsection, we evaluate the robustness of the proposed AROT. We observed success ratio with respect to increase of sink speeds (5 - 30) at different failure node rates (5% - 20%). Randomly selected nodes send data to its uplink (parent).

In the end, each source data is delivered to the sink along the AROT. Ideally, the success ratio for end-to-end success ratio of AROT is almost 100%. However, some links on the tree will be broken by various reasons, such as node failures, obstacles, and so on. To cope well with these failures, AROT utilizes simple but robust self-recovery. Figure 6 illustrates the success ratio of AROT in various percentages of node failure. As shown in fig. 6, self recovery function of AROT can perform more than 85 % success rate even in 20% of failure nodes. This is because all the links maintain the fresh connectivity to the sink. So, the partitioned tree by failures can be fast recovered in local area.

5.3 Communication overhead

In this subsection, the communication overhead (CO) in the network with respect to sink mobility is evaluated. Communication overhead rate is defined as the ratio of the number of working nodes, participating in the update reversal, over the total number of nodes.

First, the CO at each update rate from 5 seconds to 20 seconds with respect to sink movement (to each specific moving distance) is observed.

In the second experiment, the Average CO at different sink speeds, from 5m/s to 30m/s, is examined respectively.

As presented in fig.7, the simulation results illustrate that the presented AROT enables to maintain the dissemination path only with the number of working nodes less than approximately 14% of all nodes in the network, even in highly mobile sink environments.

5.4 Evaluation of End-to-end Delay

In this subsection, end-to-end delay is evaluated. We observed the end-to-end delay (from source to sink) with respect to increase of sink speeds (5 - 20) at different update reversal rates (5 - 20 seconds). Randomly selected nodes send data to its uplink (parent).

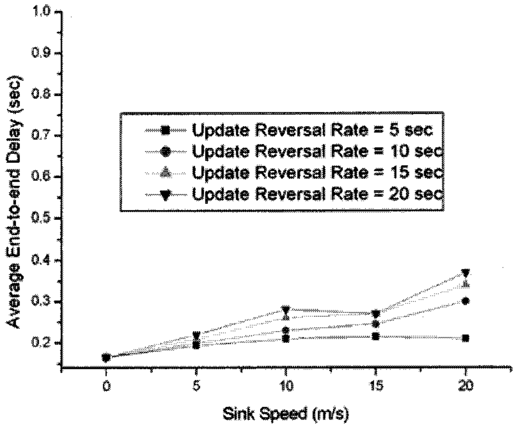
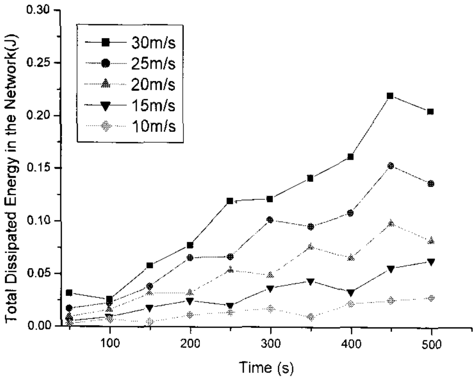
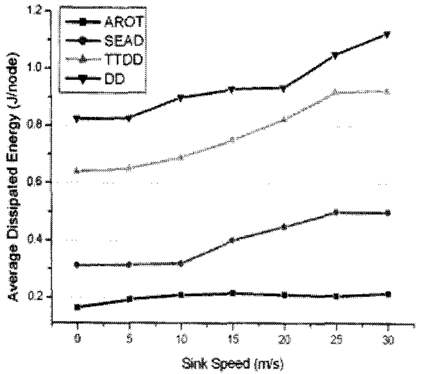


Fig. 8 Average End-to-end delay vs. Sink speed



(a) Energy consumption v.s time



(a) Energy consumption v.s. Sink Speed

Fig. 9 Evaluations of energy efficiency

We measured the total time spent in delivering data to sink and averaged the delay. In addition, each source generates event data with data update rate smaller than reversal rate. Data is generally disseminated upwardly along the tree. However, during the time when link is not stable, some nodes must wait until the next reversal process by sink is completed. Figure 8 shows the variation of average end-to-end delay with respect to increase of sink speed at different reversal rate. It is demonstrated that even if delay is more increased at less frequent reversal rate, since the tree keeps the optimal path, overall end-to-end delay with respect to the sink speed is gradually increased.

5.5 Energy consumption

In this subsection, energy consumption is evaluated. The total dissipated energy at each sink speed (from 10m/s to 30m/s) is observed over time and compared to other data dissemination protocols, directed diffusion, TTDD, and SEAD. It is remarkable that the presented AROT outperforms other protocols considerably in terms of total dissipated energy as shown in fig.9. This indicates that the AROT is able to maintain the dynamic tree with minimum communication overhead in the network.

Ⅵ. Conclusion

In this paper, in order to cope with mobile sink environments, an *Adaptive Reversal Tree* (AROT) protocol is proposed. The AROT is able to maintain the dynamic tree by adaptively reversing links with minimum communication overheads in the network, in spite of a highly mobile sink environment. Since the AROT proactively performs adaptive sink mobility management, data dissemination from each source to a mobile sink can be easily achieved along the AROT, without additional control overhead. In addition, the dynamic path is optimal in terms of hop counts and the AROT can maintain a robust tree structure by quickly recovering the partitioned tree with minimum packet transmission. Finally, the simulation results demonstrate that the AROT is a considerably energy-efficient and robust protocol.

References

〔1〕 E. M. Gafni and D. P. Bertsekas, "Distributed Algorithms for Generating Loop-Free Routes in Networks with Frequently Changing Topology," *IEEE Transactions on Communications*, January, 1981.

〔2〕 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.

〔3〕 1. C. Intanagonwiwat, R. Govindan, and

D. Estrin "Directed diffusion for Wireless Sensor Networking," *IEEE/ACM Transaction on Networking*, Vol. 11, 2003.

〔4〕 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.

〔5〕 1. 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.

〔6〕 1. Kwang-il Hwang, Jeongsik In, and Doo-seop Eom, "Dynamic Shared Tree for Minimum Energy Data Aggregation of Multiple Mobile Sinks in Wireless Sensor Networks," *EWSN2006, Lecture Notes in Computer Sciences 3868*, Zurich, Switzerland, 2006.

황 광 일 (Kwang-il Hwang) 종신회원

2002년 홍익대학교 전자 전기공학부 학사

2005년 고려대학교 전자 컴퓨터공학과 석박사통합과정 수료

2007년 고려대학교 전자컴퓨터공학과 박사

현재 시립인천전문대학 컴퓨터 제어과 전임강사

<관심분야> AD HOC networks, RFID systems, Sensor networks, Ubiquitous networks