

Routing Algorithm that Broadcasts Bitmap with Random Waiting for Wireless Sensor Networks

SANGJOON JUNG

*School of Computer Engineering
Kyungil University
Gyeongbuk, 712-701 Republic of Korea*

This paper describes a routing algorithm that broadcasts a bitmap with random waiting to reduce the number of messages transferred when establishing routing paths. Existing routing algorithms do not include routing tables for determining a path, as this would increase the cost, since sensor nodes are invariably mobile and have a limited capacity and restricted resources. However, this study proposes a sensor node that includes a routing table in the form of a bitmap, thereby reducing the size of the routing table. A bitmap is formed with a two-dimensional array, where each row and column are represented by one bit. Plus, when a node transfers routing information, it must wait randomly before transmitting to its neighbors, allowing redundant paths to open up when the bitmap is broadcast. After broadcasting, if a node then receives a bitmap from other adjacent nodes, it only updates its own bitmap. Thus, each node has a bitmap with partial link information on the network, rather than total link information. As a result, the proposed routing algorithm effectively reduces the number of messages involved in routing establishment when compared to existing algorithms.

Keywords: wireless sensor network, routing algorithm, routing protocol, bitmap broadcast, routing table, random waiting

1. INTRODUCTION

A sensor network involves a large number of tiny sensing devices that are scattered and deployed throughout the region of interest. Such sensors are generally equipped with data processing and communication capabilities, allowing them to measure ambient conditions related to the environment surrounding the sensor and deliver reports to a specific region or base station [1-4]. The base station then aggregates and analyzes the reported messages through an established path. Yet, for efficient data aggregation, the routing in sensor networks is an essential consideration, as sensor nodes are constrained in terms of their energy supply, transmission power, processing capacity, and communication abilities [5-8].

To overcome such limitations in sensor networks, various types of routing algorithm have already been proposed to establish a routing path [9-15], and can be generally classified as data-centric algorithms, hierarchical algorithms, or location-based algorithms. In data-centric algorithms, the sink sends queries to certain regions, then waits for data from the sensors located in the selected regions. Yet, since the data is requested based on queries, attribute-based naming is necessary to specify the properties of the requested data [11, 12]. Meanwhile, in hierarchical algorithms, as a single-tier network causes a gate-

Received September 15, 2006; accepted February 6, 2007.
Communicated by Ten H. Lai, Chung-Ta King and Jehn-Ruey Jiang.

way to overload with too much data aggregation, network clustering has been proposed to improve the system without any additional load or degradation of the service [13, 14]. Finally, location-based algorithms use geographically informed neighbor selection heuristics to route a packet towards the target region [15]. Yet, since the path is established based on querying data from a sink node to transfer the sensed data from the nodes, establishing and maintaining a path is the most time-consuming task. Thus, a new routing algorithm is necessary to reduce the overload in the network.

In most current sensor network environments, the sensor nodes are stationary, while the sink supports mobility [5-9]. However, building a network where each sensor node has a routing table, yet the size of the routing tables is minimized is difficult. Accordingly, a new routing algorithm is proposed that uses a bitmap and random waiting before broadcasting to neighbor nodes. Thus, each node has a routing table, which is expressed as a bitmap that contains link information. Plus, after broadcasting, a node only updates its own bitmap if it receives a bitmap from other adjacent nodes. Therefore, each node has a bitmap with partial link information on the network rather than total link information. As a result, the proposed routing algorithm reduces the number of messages involved in routing establishment when compared to existing algorithms.

2. RELATED WORKS

The process of establishing a route from sensor nodes to a sink has already received a lot of research attention. As a result, the major routing protocols for wireless sensor networks fall under the categories of data-centric, hierarchical, or location-based algorithms [5]. The following provides a brief description of an example of each type of routing algorithm.

2.1 Data-centric Algorithms

Directed diffusion [11, 12] is a data-centric routing algorithm that uses a naming scheme for data. Data-centric algorithms are query-based and depend on naming the requested data, thereby helping to eliminate many redundant transmissions. In directed diffusion, a base station called a sink node sends an interest, which is a task description, to all nodes. The task description is an interest that is named by assigning attribute-value pairs that describe the task [5-9]. The sink then sends queries to certain regions and waits for data from the sensors located in the selected regions [5, 10]. Nodes able to satisfy the interest then disseminate information along the reverse path of the interest propagation. On receiving the same interest from multiple neighboring nodes, the node may propagate events along the corresponding multiple links. Although interests initially specify a low rate of data flow, when the sink starts receiving events it will reinforce one neighbor by requesting higher data rate events. This process proceeds recursively until it reaches the nodes generating the events, causing them to generate events at a higher data rate [11, 12].

Directed diffusion is an effective method that does not require a node addressing mechanism, plus each node can perform aggregation, caching, and sensing without the need for a global network topology. However, directed diffusion cannot be applied to

certain applications, as it is based on a query-driven data delivery model. Thus, applications that request continuous monitoring data from the sink will not work efficiently.

2.2 Hierarchical Algorithms

Hierarchical algorithms focus on clustering nodes so that the cluster heads can aggregate and reduce data to save energy. However, a single-tier network can cause a gateway to overload with too much data aggregation, resulting in communication latency and inadequate event tracking. Thus, to improve such schemes without any additional load or degradation in service, network clustering has been proposed [5, 13, 14].

The low-energy adaptive clustering hierarchy (LEACH) algorithm [13], one of the most popular hierarchical routing algorithms for sensor networks, forms clusters of sensor nodes based on the received signal strength, then uses the local cluster heads as routers to the sink. The computation is also performed locally in each cluster to reduce the amount of data transmitted to the sink, plus the energy dissipation is significantly reduced, as the transmissions are only performed by the cluster heads rather than all the sensor nodes.

Furthermore, the LEACH algorithm enables the energy dissipation to be evenly distributed among the sensors, as the cluster heads are randomly selected, thereby increasing the lifetime of the network. However, this dynamic clustering also involves additional energy consumption for head changing and head advertisement.

2.3 Location-based Algorithms

In sensor networks, location information is used so that sensor nodes can report to the sink, while minimizing their energy consumption. In addition, sensor nodes calculate the distance between their neighbors to report and route, while sensor locations can be used to diffuse a query to a particular region or node, thereby reducing the number of transmissions. As such, various algorithms have been proposed based on these concepts.

The geographic and energy aware routing (GEAR) [15] technique uses energy aware and geographically informed neighbor selection heuristics to route a packet towards a target region. Within a region, it then uses a recursive geographic forwarding technique to disseminate the packet. In GEAR, each node keeps an estimated cost based on its learning distance to the destination through its neighbors [15].

Consequently, geographic routing essentially attempts to improve efficiency relative to flooding data. However, many location-based algorithms are not applicable to sensor networks where there is mobility.

3. ROUTING ALGORITHM WITH RANDOM WAITING BEFORE BROADCASTING

The next section presents the proposed routing algorithm that uses random waiting before broadcasting a routing table. The routing table only contains neighbor links to minimize its size. The following describes how a sensor node broadcasts a bitmap and why it waits randomly.

3.1 Bitmap Routing Table

A sensor node is designed with a routing table that consists of a bitmap. To reduce the power consumption, the number of transmitted messages is restricted based on all nodes keeping a routing table to configure the destination. Plus, all nodes execute the same process as follows.

- Each node only broadcasts its routing table once.
- Each node only updates its own routing table without broadcasting.
- Each node must wait a short time at random before propagating.

The reason for random waiting is described in section 4.1. Meanwhile, a bitmap is used to reduce the number of messages required to establish the paths from the nodes to a sink. When a routing table is used in a sensor network, the size of the table should be minimized, plus a routing table can only be included in a sensor node if the node is assumed to be immobile. If a node contains routing information, the number of messages between the sink and nodes can also be reduced. Fig. 1 shows the structure of a bitmap.

$$Node_ID[i] = [id_0 \quad id_1 \quad \dots \quad id_i \quad \dots \quad id_{n-1}]$$

$$link[i][j] = \begin{bmatrix} 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix}$$

Fig. 1. Structure of bitmap.

Each sensor node has two arrays that contain node *ids* and link information, respectively. The node *id* array indicates an adjacent node and the link array shows the link of that node. As such, the node *id* array includes *i*th information represented *i*th adjacent node and related to the *i*th bit of the adjacent node in the node *id* array. The position of the *i*th bit in the node *id* array corresponds to the *i*th row and *j*th column in the link array.

3.2 Routing Algorithm using Random Waiting Before Broadcasting

To compose a route using a bitmap in a node, the sink broadcasts an interest, then each sensor node determines the attributes of its sensing and broadcasts its bitmap to its neighbors. The next section shows the architecture of the transmitted messages, the routing algorithm used to update a bitmap, and bitmap propagation process.

3.2.1 Structure of interest and propagation of bitmap

When transmitting an interest to neighbor nodes, the interest is defined as follows:

```

type = four-legged animal      // target task
interval = 20ms                // send back events each 20ms
duration = 10 seconds          // for next 10 seconds
rect = [-100, 100, 200, 400]  // target location
sink node id = 0x01           // sink node id is 1

```

The type of interest is the same as with the directed diffusion algorithm, yet the sink node *id* is added in the last record to expand the sensing area.

As the number of sensor nodes increases, the size of the routing bitmap inevitably grows in spite of expressing the bitmap by bits. To apply a routing algorithm using a routing table in a large sensor field, the routing table must maintain a constant size. Thus, when creating a bitmap at the beginning of broadcasting, the number of adjacent nodes in the bitmap is limited to n . If the number of adjacent nodes in the bitmap is then larger than n , the node broadcasts an interest where the sink node *id* in the interest is substituted with its own node *id*. As a result, a node that receives the interest containing the substituted node *id* regards the propagating node as the sink node. The node then makes a bitmap in the same way as the neighbor nodes of the original sink node.

If a node has n number of neighbors in the bitmap, the node changes the sink node *id* to its own node *id*. For example, if the applied number of nodes is 30, this node then broadcasts an interest where the sink node *id* has been changed as follows. Thus, each node maintains a constant number of nodes in its bitmap.

```

type = four-legged animal      // target tasks
interval = 20ms                // send back events each 20ms
duration = 10 seconds          // for next 10 seconds
rect = [-100, 100, 200, 400]  // target location
sink node id = 0x1E           // sink node id is 30

```

The next nodes then create a new bitmap where the sink node is node 30 and report an event by identifying the sink node as node 30. As a result, the proposed algorithm maintains a constant size for the bitmap, while expanding the sensing area.

3.2.2 Routing algorithm

Each node investigates whether the attributes of the interest match its sensing tasks. If the node does not find a suitable match, it then broadcasts the received interest and its own bitmap. The updating mechanism occurs when a node receives an interest from its adjacent node. Creating a bitmap consists of four steps as follows.

Step 1: When a node first receives a broadcast message from its neighbor, the node determines whether its data matches the attributes in the message by identifying the interest.

Step 2: The node updates its own bitmap as follows:

- if (the received node *id* does not exist)
 1. Insert node *id* at the *i*th-position in the node *id* array.
 2. Set 1 into the *i* and *j*th-position in the link array.

```

else
  {bitmap = own bitmap || received bitmap
   where || denotes or-operation.}
end if
    
```

Step 3: The node must wait a short time randomly before broadcasting its bitmap information to its neighbors.

Step 4: If the node receives other request messages from adjacent nodes, it then just updates its own bitmap information without propagation.

In step 3, a node that receives an interest and bitmap from an adjacent node must wait randomly before broadcasting, allowing redundant paths to open up in the bitmap. In a MAC layer, a collision occurs when two nodes try to transmit a signal at the same time. In this case, the nodes then wait randomly and transmit again. Therefore, this collision-avoidance mechanism is applied to the proposed algorithm, so all nodes must wait randomly before transmitting. After the timer expires, the nodes send an interest and bitmap, resulting in abundant paths in the bitmap.

After an interest is broadcast, each node then has a bitmap that contains link information about its adjacent nodes. Yet, the bitmap is different in each node, as the routing messages are only broadcast once. Thus, each node only has partial information on the network topology.

3.2.3 Propagation of bitmap

Each node tries to broadcast a bitmap to its neighbors. The nodes must then wait randomly before broadcasting. Fig. 2 shows the routing steps and configuration of a bitmap when the proposed routing algorithm is executed.

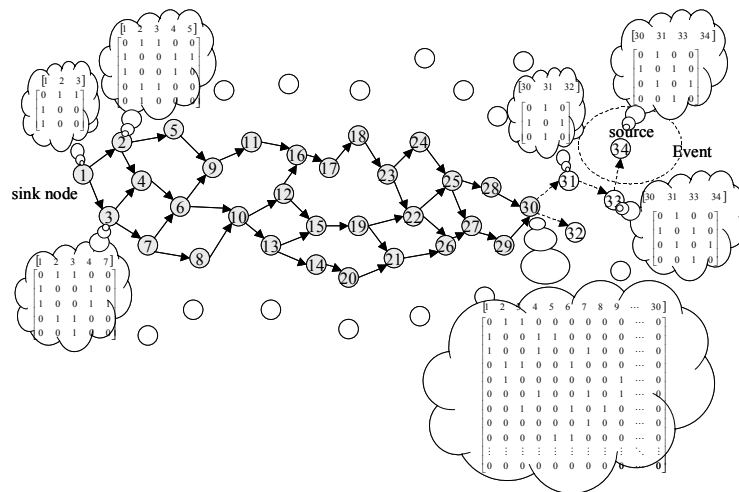


Fig. 2. Propagation of bitmap.

When using the proposed algorithm, in the first step, a sink sends its bitmap to its neighbors to compose a routing table. The adjacent nodes keep the bitmap from node 1, then wait randomly before broadcasting their routing table, thereby opening up redundant paths. A detailed explanation is given in section 4.1. Nodes 2, 3, and 4 operate their timer. As soon as a node receives a bitmap from its neighbor, it checks the interest and determines the waiting time interval before broadcasting. After broadcasting, a node only updates its own bitmap without any propagation. By doing this repeatedly, each node keeps a bitmap that contains partial link information about the whole network. In node 30, as the interest does not match its sensing tasks, the node determines to change the sink node id in the interest and propagates it to the adjacent nodes. The adjacent nodes then operate in the same way as the neighbors of the original sink node, as such, nodes 31, 32, 33, and 34 receive the interest that describes the sink node as node 30. Consequently, all the sensor nodes participating in the sensor field have a bitmap when the routing algorithm is executed.

After completing the proposed routing algorithm, the bitmaps are slightly different in each node. Thus, a sensor node involved in the requested sensing tasks identifies a sink node and path by investigating its bitmap. All nodes can find a proper path to reach the sink using the BFS (Breath first search) Algorithm.

4. ANALYSIS

The proposed algorithm is compared with three idealized schemes for data dissemination, including the directed diffusion algorithm, LEACH, and GEAR.

4.1 Comparison of Broadcasting When using Flooding and Random Waiting

With the proposed algorithm, when a node receives an interest, it can include redundant paths based on waiting randomly before broadcasting. However, there are two methods of broadcasting a bitmap when a sensor node propagates a bitmap to its neighbors: flooding and random waiting, and Fig. 3 shows a comparison of the two methods.

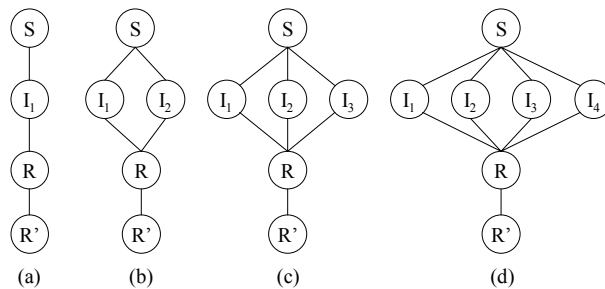


Fig. 3. Comparison of broadcasting when using flooding and random waiting.

In the case of flooding, node R broadcasts its bitmap to node R' immediately before receiving information from the other adjacent nodes. Thus, node R reports its bitmap to

node R' without any information from the other intermediate nodes. When the number of intermediate nodes is one, there is no difference between flooding and random waiting (Fig. 3 (a)). If node R operates on a random timer, it has a chance to receive information from node I with adjacent links (Figs. 3 (b), (c), and (d)), then node R can report to node R' and include redundant paths in its bitmap. Table 1 shows the number of links in a bitmap.

Table 1. Number of links in bitmap.

Broadcasting characteristics	Worst case	Average case	Best case
With flooding	1	1	1
With random waiting	1	$k/2$	k



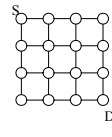
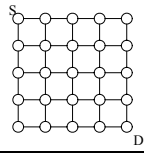
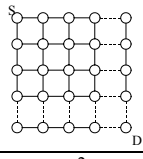
k : number of intermediate nodes reporting to next node

When broadcasting with random waiting, a sensor node can include more redundant paths in its bitmap. Thus, if the waiting time at node R is greater than that for all the intermediate nodes, node R' will receive the complete link information.

4.2 Number of Transmitted Messages

To evaluate the efficiency of routing algorithms for sensor networks, the number of messages needs to be compared. Thus, to determine the number of messages involved in configuring a bitmap, the evaluated topologies were generalized as shown in Table 2.

Table 2. Generalized topologies for evaluating messages.

	2 * 2	3 * 3	4 * 4	5 * 5	$n * n$
Topology					
Number of nodes	4	9	16	25	n^2
Number of links (l)	4	12	24	40	$2n(n-1)$
Number of hops	2	4	6	8	$2n-2$
Number of clusters	0	3	4	7	$\left\lceil \frac{n}{2} \right\rceil^2$

A simulation topology is usually defined as a free domain. Therefore, to obtain a regular result for deriving the characteristics, square topologies were defined by 2 * 2, 3 * 3, 4 * 4, 5 * 5, and $n * n$. Table 3 shows a comparison of the proposed algorithm and existing routing algorithms. The number of cluster is calculated to compare to the hierarchical protocols. The method of calculating clusters is follows in [16].

The number of messages was determined based on the number sent and the number

Table 3. Number of broadcast messages.

Item Scheme	Directed Diffusion	LEACH	GEAR	Proposed Algorithm
Messages sent	$n + n = 2n$	$n + \left\lceil \frac{n}{2} \right\rceil$	$2n - 2 + n^2 / \left\lceil \frac{n}{2} \right\rceil^2$	n
Messages Received	$2 * l + n$ $= 4n^2 - 3n$	$2l + 2 \left\lceil \frac{n}{2} \right\rceil^2$	$2n^2 / \left\lceil \frac{n}{2} \right\rceil^2$	$2 * l = 4n^2 - 4n$
Total number of messages	$4n^2 - n$	$4n - 2n + 3 \left\lceil \frac{n}{2} \right\rceil$	$2n - 2 + 3n^2 / \left\lceil \frac{n}{2} \right\rceil^2$	$4n^2 - 3n$

n : number of nodes, l : number of links

received by a node. With directed diffusion, each node sends an interest when the task is configured. When a target node matches a sensing task, the intermediate nodes also propagate to reinforce the path a second time. As a result, messages were sent $2n$ times. Meanwhile, since all nodes broadcast messages to their neighbors, the number of received messages was defined by adding a double link to the total number of nodes. For LEACH and GEAR, clustering forms are adopted, so the total number of messages was calculated by adding the number of links and clusters.

The number of total messages with the proposed algorithm was smaller than that with the existing routing algorithms, as shown in Table 3, as the routing algorithm only broadcasts messages once based on random waiting, thereby reducing the power and conserving the resources.

5. CONCLUSIONS

This paper proposed a routing algorithm for a sensor network, where a bitmap is broadcast using random waiting at a sensor node. As the main focus of the proposed algorithm is to reduce the battery power involved in establishing routing paths by decreasing the number of messages, path selection is optimized using a routing table based on bitmaps. As such, each node has a bitmap that contains link information about the neighbor nodes. Upon receiving a bitmap, each node then waits randomly before broadcasting the bitmap to its neighbors. After broadcasting, if a node then receives a bitmap from other adjacent nodes, the node only updates its own bitmap. Thus, when using the proposed algorithm, each node includes routing information to find a destination node for transferring packets. It is not important for each node to have the same routing bitmap, as each node knows the adjacent nodes and paths that have been triggered by the sink. Each node then caches its bitmap until information on a distinct interest is received. Future research on the proposed algorithm will involve implementation in a sensor node.

REFERENCES

1. D. Tian and N. D. Georganas, "Energy efficient routing with guaranteed delivery in

- wireless sensor networks,” *IEEE Wireless Communications and Networking*, Vol. 3, 2003, pp. 1923-1929.
2. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, Vol. 38, 2002, pp. 393-422.
 3. K. Sohrahi, J. Gao, V. Ailawadhi, and G. J. Pottie, “Protocols for self-organization of a wireless sensor network,” *IEEE Personal Communications*, Vol. 7, 2000, pp. 16-27.
 4. R. Min, M. Bhardwaj, S. Cho, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, “Low-power wireless sensor networks,” in *Proceedings of International Conference on VLSI Design*, 2001, pp. 205-210.
 5. K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Elsevier Ad Hoc Network Journal*, Vol. 3, 2005, pp. 325-349.
 6. V. Rodoplu and T. H. Meng, “Minimum energy mobile wireless networks,” *IEEE Journal of Selected Areas in Communication*, Vol. 17, 1999, pp. 1333-1344.
 7. L. Li and J. Halpern, “Minimum-energy mobile wireless networks revisited,” in *Proceedings of IEEE Conference of Communications*, 2001, pp. 278-283.
 8. R. Krishnan and D. Starobinski, “Efficient clustering algorithms for self-organizing wireless sensor networks,” *Journal of Ad-Hoc Networks*, Vol. 4, 2006, pp. 36-59.
 9. S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, “A taxonomy of wireless microsensor network models,” *Mobile Computing and Communications Review*, Vol. 6, 2002, pp. 28-36.
 10. W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 174-185.
 11. C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, 2000, pp. 56-67.
 12. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: scalable coordination in sensor networks,” in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 263-270.
 13. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless sensor networks,” in *Proceedings of the Hawaii International Conference System Sciences*, 2000, pp. 3005-3014.
 14. L. Subramanian and R. H. Katz, “An architecture for building self configurable systems,” in *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing*, 2000, pp. 63-73.
 15. Y. Yu, D. Estrin, and R. Govindan, “Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks,” Technical Report No. UCLA/CSD-TR-01-0023, Department of Computer Science, University of California, Los Angeles, 2001.
 16. C. R. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal on Selected Areas in Communications*, Vol. 15, 1997, pp. 1265-1275.



Sangjoon Jung received the M.S. and Ph.D. in Department of Computer Engineering from the Yeungnam University in 2001 and 2005. In 2003, he joined the School of Computer Engineering at Kyungil University, where he is currently a full-time instructor. His research interests include wireless sensor networks, ad hoc networks, networks management, and traffic engineering.